

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2000134242 A**

(43) Date of publication of application: 12 . 05 . 00

(51) Int. Cl.

**H04L 12/40**  
**G06F 13/362**(21) Application number: **10321490**

(22) Date of filing: 27 . 10 . 98

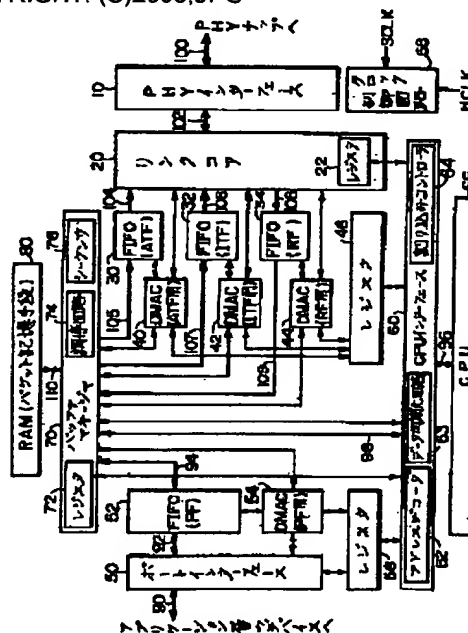
(71) Applicant: **SEIKO EPSON CORP**(72) Inventor: **HORIUCHI HIROSHI**  
**KANBARA YOSHIYUKI****(54) DATA TRANSFER CONTROLLER AND ELECTRONIC EQUIPMENT****(57) Abstract:**

**PROBLEM TO BE SOLVED:** To provide a data transfer controller and electronic equipment capable of reducing the overheads of processings and transferring data at a high speed with the hardware of a small scale.

**SOLUTION:** This data transfer controller of IEEE1394 is provided with first buses 90, 92 and 94, second buses 96 and 98, third buses 100-109 and a forth bus 110 electrically connected to the application of a poststage, a CPU 66, a PHY chip and a RAM 80. A mediation circuit 74 performs mediation for establishing a data route between one of the first, second and third buses and the forth bus. DMACs 54, 40, 42 and 44 for transferring the data without the intervention of the CPU 66 and FIFOs 30, 32 and 34 for phase adjustment are provided and the stage number of the FIFO is made equal to or less than three. The RAM is separated into a header area and a data area and the header area and the data area are separated into areas for reception and for transmission. By using TAG, the header of a reception packet is written in the header area and the data are written in the data area. The header is read from the header area, the data are read from the data area and a transmission packet is

assembled.

COPYRIGHT: (C)2000,JPO





## 【特許請求の範囲】

【請求項 1】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、後段のアプリケーションに接続される第 1 のバスと、該データ転送制御装置をコントロールする第 2 のバスと、

物理層のデバイスに電氣的に接続される第 3 のバスと、ランダムアクセスが可能でありパケットを格納するための記憶手段に電氣的に接続される第 4 のバスと、前記第 1、第 2、第 3 のバスのいずれかと前記第 4 のバスとの間にデータ経路を確立するための調停を行う調停手段とを含むことを特徴とするデータ転送制御装置。

【請求項 2】 請求項 1 において、前記アプリケーションと前記記憶手段との間でのデータ転送を前記第 2 のバスからの介入なしに行うための第 1 の DMA コントローラと、前記物理層のデバイスと前記記憶手段との間でのデータ転送を前記第 2 のバスからの介入なしに行うための第 2 の DMA コントローラとを含むことを特徴とするデータ転送制御装置。

【請求項 3】 請求項 1 又は 2 において、前記物理層のデバイスと前記記憶手段との間でのデータ転送のための位相調整用の FIFO を含むことを特徴とするデータ転送制御装置。

【請求項 4】 請求項 3 において、前記調停手段が、前記物理層のデバイスと前記記憶手段との間でのデータ転送の優先順位を最も高くし、前記 FIFO の段数が 3 段以下であることを特徴とするデータ転送制御装置。

【請求項 5】 請求項 1 乃至 4 のいずれかにおいて、前記記憶手段が、パケットの制御情報が格納される制御情報領域と、パケットのデータが格納されるデータ領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 6】 請求項 5 において、前記記憶手段の前記制御情報領域が、受信制御情報領域と送信制御情報領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 7】 請求項 5 又は 6 において、前記記憶手段の前記データ領域が、受信データ領域と送信データ領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 8】 請求項 5 乃至 7 のいずれかにおいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むパケット分離手段を含むことを特徴とするデータ転送制御装置。

【請求項 9】 請求項 8 において、パケットの少なくとも制御情報とデータを区別するためのタグ情報を生成すると共に、生成された該タグ情報を

パケットに関連づける手段を含み、前記パケット分離手段が、パケットに関連づけられた前記タグ情報に基づいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むことを特徴とするデータ転送制御装置。

【請求項 10】 請求項 5 乃至 9 のいずれかにおいて、パケットの制御情報を前記記憶手段の前記制御情報領域から読み出し、該制御情報と対をなすパケットのデータを前記記憶手段の前記データ領域から読み出すパケット結合手段を含むことを含むことを特徴とするデータ転送制御装置。

【請求項 11】 請求項 10 において、前記パケット結合手段が、前記データ領域から読み出されるデータのアドレスを指すデータポインタを、前記制御情報領域から読み出された制御情報から取得し、取得したデータポインタを用いて前記データ領域からデータを読み出すことを特徴とするデータ転送制御装置。

【請求項 12】 請求項 1 乃至 11 のいずれかにおいて、IEEE 1394 の規格に準拠したデータ転送を行うことを特徴とするデータ転送制御装置。

【請求項 13】 請求項 1 乃至 12 のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする電子機器。

【請求項 14】 請求項 1 乃至 12 のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、処理が施されるデータを取り込むための装置とを含むことを特徴とする電子機器。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データ転送制御装置及びこれを含む電子機器に関する。

【0002】

【背景技術及び発明が解決しようとする課題】 近年、IEEE 1394 と呼ばれるインターフェース規格が脚光を浴びている。この IEEE 1394 は、次世代のマルチメディアにも対応可能な高速シリアルバスインターフェースを規格化したものである。この IEEE 1394 によれば、動画像などのリアルタイム性が要求されるデータも扱うことができる。また、IEEE 1394 のバスには、プリンタ、スキャナ、CD-R ドライブ、ハードディスクドライブなどのコンピュータの周辺機器のみならず、ビデオカメラ、VTR、TV などの家庭用電化

製品も接続できる。このため、電子機器のデジタル化を飛躍的に促進できるものとして期待されている。

【0003】このようなIEEE1394の概要については、例えば「IEEE1394ハイ・パフォーマンス・シリアルバスの概要」(Interface Apr. 1996の1~10頁)、「PC周辺機器用バス規格群総覧」(Interface Jan. 1997の106頁~116頁)、「IEEE1394-1995(FireWire)のリアルタイム転送モードとマルチメディア対応プロトコル」(Interface Jan. 1997の136~146頁)に開示されている。また、IEEE1394に準拠したデータ転送制御装置としては、テキサス・インスツルメンツ社製のTSB12LV31などが知られている。

【0004】しかしながら、このIEEE1394に準拠したデータ転送制御装置には次のような課題があることが判明した。

【0005】即ち、現在のIEEE1394規格によれば最大で400Mbpsの転送速度が実現可能となっている。しかし、現実には、処理のオーバーヘッドの存在に起因して、システム全体の実転送速度はこれよりもかなり低くなっている。つまり、CPU上で動作するファームウェアやアプリケーションソフトが、送信データを準備したり、受信データを取り込んだりするなどの処理に多くの時間を要してしまい、バス上での転送速度が速くても、結局、高速なデータ転送を実現できない。

【0006】特に、周辺機器に組み込まれるCPUは、パーソナルコンピュータなどのホストシステムに組み込まれるCPUに比べて処理能力が低い。このため、ファームウェアやアプリケーションソフトの処理のオーバーヘッドの問題は、非常に深刻なものとなる。従って、このようなオーバーヘッドの問題を効果的に解消できる技術が望まれている。

【0007】本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、ファームウェアやアプリケーションソフトなどの処理のオーバーヘッドを軽減し、小規模なハードウェアで高速なデータ転送を実現できるデータ転送制御装置及びこれを用いられる電子機器を提供することにある。

【0008】

【課題を解決するための手段】上記課題を解決するために本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、後段のアプリケーションに接続される第1のバスと、該データ転送制御装置をコントロールする第2のバスと、物理層のデバイスに電気的に接続される第3のバスと、ランダムアクセスが可能でありパケットを格納するための記憶手段に電気的に接続される第4のバスと、前記第1、第2、第3のバスのいずれかと前記第4のバスとの間にデータ経路を確立するための調停を行う調停手段とを含む

ことを特徴とする。

【0009】本発明によれば、互いに分離される第1、第2、第3のバスが設けられる。そして調停手段の調停により、これらの第1、第2、第3のバスのいずれかと、ランダムアクセス可能な記憶手段の第4のバスとの間にデータ経路が確立される。このようにすることで、他のノードから物理層のデバイスを介して受信したパケットを、所望の配置で記憶手段に格納することが可能になる。また、パケットの制御情報の読み出しや書き込みは第2のバスを用いて行い、パケットのデータの読み出しや書き込みは第1のバスを用いて行うことも可能になる。これにより、トランザクション層やアプリケーション層などの上層の処理負担を軽減できる。また、第1、第2のバスとして低速なバスを採用したり、データ転送制御装置をコントロールするデバイスとして低速で安価なデバイスを採用することができる。この結果、データ転送制御装置の小規模化、低コスト化を図れるようになる。

【0010】なお、第1、第2、第3、第4のバスは、各々、アプリケーション、データ転送制御装置をコントロールするデバイス、物理層のデバイス、RAMに、少なくとも電気的に接続されていればよく、これらのバス上に他のデバイスを介在させてもよい。

【0011】また本発明は、前記アプリケーションと前記記憶手段との間でのデータ転送を前記第2のバスからの介入なしに行うための第1のDMAコントローラと、前記物理層のデバイスと前記記憶手段との間でのデータ転送を前記第2のバスからの介入なしに行うための第2のDMAコントローラとを含むことを特徴とする。このようにすることで、アプリケーションと記憶手段との間や、物理層のデバイスと記憶手段との間でデータ転送を行う際に、データ転送制御装置をコントロールするデバイスが介入しなくてもよくなる。この結果、データ転送制御装置をコントロールするデバイスの処理負担を大幅に軽減できる。

【0012】また本発明は、前記物理層のデバイスと前記記憶手段との間でのデータ転送のための位相調整用のFIFOを含むことを特徴とする。このようにすれば、調停手段による調停があった場合にも、物理層のデバイスと記憶手段との間でのデータ転送を適正に行うことが可能になる。

【0013】また本発明は、前記調停手段が、前記物理層のデバイスと前記記憶手段との間でのデータ転送の優先順位を最も高くし、前記FIFOの段数が3段以下であることを特徴とする。このようにすれば、物理層のデバイスと記憶手段との間でのデータ転送を優先して行った場合にも、データが失われてしまうなどの事態を防止できる。

【0014】また本発明は、前記記憶手段が、パケットの制御情報が格納される制御情報領域と、パケットのデ

ータが格納されるデータ領域とに分離されていることを特徴とする。このようにすることで、トランザクション層やアプリケーション層などの上層の処理負荷を軽減でき、システム全体の実転送速度を向上できる。また、記憶手段からのパケットの読み出し処理や記憶手段への書き込み処理を簡素化できる。

【0015】また本発明は、前記記憶手段の前記制御情報領域が、受信用制御情報領域と送信用制御情報領域とに分離されていることを特徴とする。このようにすることで、受信用制御情報領域から制御情報を連続して読み出ししたり、送信用制御情報領域に制御情報を連続して書き込むことが可能になり、処理の簡素化、処理負担の軽減を図れる。

【0016】また本発明は、前記記憶手段の前記データ領域が、受信用データ領域と送信用データ領域とに分離されていることを特徴とする。このようにすることで、受信用データ領域から連続して受信データを読み出ししたり、送信用データ領域に連続して送信データを書き込むことが可能になり、上層の処理のオーバーヘッドを軽減できる。

【0017】また本発明は、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むパケット分離手段を含むことを特徴とする。このようにすれば、パケットの制御情報とデータが自動的に、各々、制御情報領域、データ領域に書き込まれるようになり、上層の処理のオーバーヘッドを軽減できる。

【0018】また本発明は、パケットの少なくとも制御情報とデータを区別するためのタグ情報を生成すると共に、生成された該タグ情報をパケットに関連づける手段を含み、前記パケット分離手段が、パケットに関連づけられた前記タグ情報に基づいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むことを特徴とする。このようにすることで、簡易なハードウェア構成で、パケットの制御情報を制御情報領域にデータをデータ領域に格納することが可能になる。

【0019】また本発明は、パケットの制御情報を前記記憶手段の前記制御情報領域から読み出し、該制御情報と対をなすパケットのデータを前記記憶手段の前記データ領域から読み出すパケット結合手段を含むことを含むことを特徴とする。このようにすれば、上層は、パケットを送信する順序に依存せずに、制御情報やデータを記憶手段に書き込める。また、上層は、送信すべきパケットのデータをデータ領域に連続して書き込むこともできる。また、制御情報領域の制御情報とデータ領域のデータとを結合してパケットを組み立てる処理に、上層が関与する必要がない。従って、上層の処理負担を格段に軽減できるようになる。

【0020】また本発明は、前記パケット結合手段が、

前記データ領域から読み出されるデータのアドレスを指すデータポインタを、前記制御情報領域から読み出された制御情報から取得し、取得したデータポインタを用いて前記データ領域からデータを読み出すことを特徴とする。このようにすれば、制御情報と対になるデータを適正にデータ領域から読み出せるようになると共に、制御情報とデータとを結合しパケットを組み立てる処理の簡素化を図れる。

【0021】なお、本発明では、IEEE1394の規格に準拠したデータ転送を行うことが望ましい。

【0022】また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする。また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、処理が施されるデータを取り込むための装置とを含むことを特徴とする。

【0023】本発明によれば、他のノードから転送されたデータを電子機器において出力したり記憶したりする処理、電子機器において取り込んだデータを他のノードに転送したりする処理を高速化することが可能になる。また、本発明によれば、データ転送制御装置を小規模化できると共に、データ転送を制御するファームウェアなどの処理負担を軽減できるため、電子機器の低コスト化、小規模化などを図ることも可能になる。

【0024】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0025】1. IEEE1394

まず、IEEE1394の概要について簡単に説明する。

【0026】1. 1 データ転送速度、接続トポロジー  
IEEE1394 (IEEE1394-1995、IEEE1394. A) では100~400Mbpsの高速なデータ転送が可能となっている (IEEE1394. Bでは800~3200Mbps)。また、転送速度が異なるノードをバスに接続することも許される。

【0027】各ノードはツリー状に接続されており、1つのバスに最大で63個のノードが接続可能になっている。なお、バスブリッジを利用すれば約64000個のノードを接続することも可能である。

【0028】電源が投入されたり、途中でデバイスの抜き差しが発生すると、バスリセットが発生し、接続トポロジーに関する情報が全てクリアされる。そして、バスリセット後、ツリー識別 (ルートノードの決定)、自己識別が行われる。その後、アイソクロナスリソースマネージャ、サイクルマスタ、バスマネージャ等の管理ノー

ドが決定される。そして、通常の packets 転送が開始される。

#### 【0029】1. 2 転送方式

IEEE1394では、packetsの転送方式として、信頼性が要求されるデータの転送に好適な非同期転送と、リアルタイム性が要求される動画や音声などのデータの転送に好適なアイソクロナス転送が用意されている。

【0030】図1(A)に、非同期サブアクションの例を示す。1つのサブアクションは、調停、packets転送、acknowledgmentからなる。即ち、データ転送に先立って、まず、busの使用権に関する調停が行われる。そしてソース(転送元)ノードからdestinationノード(転送先)ノードにpacketsが転送される。このpacketsのヘッダにはソースIDとdestinationIDが含まれる。destinationノードは、このdestinationIDを読んで、自ノード宛のpacketsか否かを判断する。destinationノードは、packetsを受け取ると、ソースノードにacknowledgment(ACK)のpacketsを返す。

【0031】packets転送とACKの間にはacknowledgmentジギャップが存在する。また、1つのサブアクションと次のサブアクションの間にはサブアクションジギャップが存在する。そして、サブアクションジギャップに相当する一定のbus・アイドル時間が経過しないと、次のサブアクションの調停を開始できない。これによりサブアクション相互の衝突が回避される。

【0032】図1(B)に、アイソクロナスサブアクションの例を示す。アイソクロナス転送はbroadcastキャスト(busに接続される全てのノードに転送)で実行されるため、packets受信時にACKは返送されない。また、アイソクロナス転送では、ノードIDではなくチャンネル番号を使用してpackets転送が行われる。なお、サブアクション間にはアイソクロナスジギャップが存在する。

【0033】図1(C)に、データ転送時のbusの様子を示す。アイソクロナス転送は、サイクルマスタが一定周期毎にサイクルスタートpacketsを発生することで開始する。これにより、1つのチャンネル当たり、125μs毎に少なくとも1つのpacketsを転送できるようになる。この結果、動画や音声などのリアルタイム性が要求されるデータの転送が可能になる。

【0034】非同期転送はアイソクロナス転送の合間に行われる。即ち、アイソクロナス転送の方が非同期転送よりも優先順位が高くなっている。これは、図1(C)に示すように、アイソクロナスジギャップの時間を、非同期転送のサブアクションジギャップの時間よりも短くすることで実現される。

#### 【0035】1. 3 ツリー識別

ツリー識別はbusリセットの後に行われる。ツリー識別によりノード間の親子関係やルートノードが決定され

る。

【0036】まず、リーフノード(1つのノードにしか接続されていないノード)が、隣接するノードに、ペアレントノティファイを送る。例えば図2(A)のようにノードA、B、C、D、Eが接続されている場合には、ノードAからノードBに、ノードD及びEからノードCに、ペアレントノティファイ(PN)が送られる。

【0037】ペアレントノティファイを受け取ったノードは、送り元のノードを自身の子と認知する。そして、チャイルドノティファイをそのノードに送る。例えば図2(A)では、ノードBからノードAに、ノードCからノードD及びEにチャイルドノティファイ(CN)が送られる。これによってノードB、A間、ノードC、D間、ノードC、E間の親子関係が決定する。

【0038】ノードB、Cの親子関係は、どちらが先にペアレントノティファイを送ったかで決定される。例えば図2(B)のように、ノードCの方が先にペアレントノティファイを送ると、ノードBが親になりノードCが子になる。

【0039】ポートの接続先の全てのノードが自身の子となるノードがルートになる。図2(B)では、ノードBがルートになる。なお、IEEE1394では、全てのノードがルートになる可能性がある。

#### 【0040】1. 4 自己識別

ツリー識別の後、自己識別が行われる。自己識別においては、接続トポロジにおいてルートノードから遠いノードから順にセルフID packetsが転送される。

【0041】より具体的には、例えば図3において、まず、ルートノードBのポート1(番号の小さいポート)に接続されるノードAが、セルフID packets(自己識別 packets)を全てのノードにbroadcastキャストする。

【0042】次に、ルートノードBのポート2(番号が大きいポート)に接続されるノードCが選択され、このノードCのポート1(番号の小さいポート)に接続されるノードDがセルフID packetsをbroadcastキャストする。次に、ノードCのポート2(番号の大きいポート)に接続されるノードEがセルフID packetsをbroadcastキャストし、その後、ノードCがbroadcastキャストする。最後に、ルートであるノードBがセルフID packetsをbroadcastキャストし、自己識別が完了する。

【0043】セルフID packetsには各ノードのIDが含まれる。broadcastキャストを行う時点で他のノードから受け取ったセルフID packetsの個数が、この各ノードのIDとなる。例えば図3では、ノードAがbroadcastキャストを行う時点では、どのノードもセルフID packetsを発していないため、ノードAのIDは0になる。ノードAは、このID=0をセルフID packetsに含ませてbroadcastキャストする。また、ノードDがbroadcastキャストを行う時点では、ノードAのみがセルフIDパ

ケットを発している。このため、ノードDのIDは1になる。同様に、ノードE、C、BのIDは、各々、2、3、4になる。

【0044】図4(A)にセルフIDパケットのフォーマットを示す。同図に示すようにセルフIDパケットには各ノードの基本情報が含まれる。具体的には、各ノードのID(PHY\_ID)、リンク層がアクティブか否か(L)、ギャップカウント(gap\_cnt)、転送速度(sp)、アイソクロナスリソースマネージャになれる能力を有するか否か(C)、電力状態(pwr)、ポートの状態(p0、p1、p2)などに関する情報が含まれる。

【0045】なお、図4(B)に、ノードのポート数が4個以上の場合に使用されるセルフIDパケット#1、#2、#3のフォーマットを示す。ポート数が4~11個の場合にはセルフIDパケット#0(図4(A))及び#1が、12~19個の場合にはセルフIDパケット#0、#1及び#2が、20~27個の場合にはセルフIDパケット#0、#1、#2及び#3が使用されることになる。

【0046】また、図4(C)、(D)に、セルフIDパケットと同様に、物理層のパケット(PHYパケット)であるリンクオンパケット、PHY構成パケットのフォーマットを示す。

【0047】1.5 アイソクロナスリソースマネージャ  
アイソクロナスリソースマネージャ(IRM)は以下の管理機能を有する。

【0048】第1に、アイソクロナス転送に必要な種々のリソースを提供する。例えば、チャネル番号レジスタや帯域幅レジスタを提供する。第2に、バスマネージャのIDを示すレジスタを提供する。第3に、バスマネージャがない場合に、簡易的なバスマネージャとなる役割を有する。

【0049】IRMになれる能力を有し(アイソクロナスリソースを管理する能力を有し)、且つ、動作状態になっている(リンク層がアクティブになっている)ノードの中で(IRMになれる資格を有するノードの中で)、ルートに最も近い(IDが最も大きい)ノードがIRMになる。より、具体的には、図4(A)のセルフIDパケットにおいて、IRMになれる能力を有するか否かを示すC(CONTENDER)ビットと、リンク層がアクティブか否かを示すL(LINK\_ACTIVE)ビットが共に1になっているノードの中で、ルートに一番近いノード(PHY\_IDが一番大きいノード)がIRMになる。例えば、ルートノードのセルフIDパケットのCビットとLビットが1の場合には、ルートノードがIRMになる。

【0050】1.6 サイクルマスタ、バスマネージャ  
サイクルマスタは、図1(C)に示すサイクルスタート

パケットを送信する役割を有し、ルートノードがサイクルマスタになる。

【0051】バスマネージャは、トポロジーマップ(各ノードの接続状態)の作成、スピードマップの作成、バスの電力管理、サイクルマスタの決定、ギャップカウンの最適化などの仕事を行う。

【0052】1.7 プロトコル構成

図5を用いて、IEEE1394のプロトコル構成(層構造)について説明する。

10 【0053】IEEE1394のプロトコルは、物理層、リンク層、トランザクション層により構成される。また、シリアルバスマネージメントは、物理層、リンク層、トランザクション層をモニターしたり制御したりするものであり、ノードの制御やバスのリソース管理のための種々の機能を提供する。

【0054】物理層は、リンク層により使用されるロジカルシンボルを電気信号に変換したり、バスの調停を行ったり、バスの物理的インターフェースを定義する。

20 【0055】リンク層は、アドレッシング、データチェック、データフレーミング、サイクル制御などを提供する。

【0056】トランザクション層は、リード、ライト、ロックなどのトランザクションを行うためのプロトコルを定義する。

【0057】物理層及びリンク層は、通常、データ転送制御装置(インターフェースチップ)などのハードウェアにより実現される。また、トランザクション層は、CPU上で動作するファームウェアや、ハードウェアにより実現される。

30 【0058】2. 全体構成

次に、本実施形態の全体構成について図6を用いて説明する。

【0059】図6において、PHYインターフェース10は、物理層のプロトコルを実現するPHYチップとのインターフェースを行う回路である。

40 【0060】リンクコア20(リンク手段)は、リンク層のプロトコルやトランザクション層のプロトコルの一部を実現する回路であり、ノード間でのパケット転送のための各種サービスを提供する。レジスタ22は、これらのプロトコルを実現したりリンクコア20を制御するためのレジスタである。

【0061】FIFO(ATF)30、FIFO(ITF)32、FIFO(RF)34は、各々、非同期送信用、アイソクロナス送信用、受信用のFIFOであり、例えばレジスタや半導体メモリなどのハードウェアにより構成される。本実施形態では、これらのFIFO30、32、34の段数は非常に少ない。例えば1つのFIFOの段数は、好ましくは3段以下であり、更に好ましくは2段以下となる。

50 【0062】DMAC40、42、44は、各々、AT



F、ITF、RF用のDMAコントローラである。これらのDMAC40、42、44を用いることで、CPU66に介入されることなく、RAM80とリンクコア20との間でのデータ転送が可能になる。なお、レジスタ46は、DMAC40、42、44などを制御するレジスタである。

【0063】ポートインターフェース50は、アプリケーション層のデバイス（例えばプリンタの印字処理を行うデバイス）とのインターフェースを行う回路である。本実施形態では、このポートインターフェース50を用いて、例えば8ビットのデータ転送が可能になっている。

【0064】FIFO(PF)52は、アプリケーション層のデバイスとの間でのデータ転送のためのFIFOであり、DMAC54は、PF用のDMAコントローラである。レジスタ56は、ポートインターフェース50やDMAC54を制御するレジスタである。

【0065】CPUインターフェース60は、データ転送制御装置をコントロールするCPU66とのインターフェースを行う回路である。CPUインターフェース60は、アドレスデコーダ62、データ同期化回路63、割り込みコントローラ64を含む。クロック制御回路68は、本実施形態で使用されるクロックを制御するものであり、PHYチップから送られてくるSCLKや、マスタークロックであるHCLKが入力される。

【0066】バッファマネージャ70は、RAM80とのインターフェースを管理する回路である。バッファマネージャ70は、バッファマネージャの制御のためのレジスタ72、RAM80へのバス接続を調停する調停回路74、各種の制御信号を生成するシーケンサ76を含む。

【0067】RAM80は、ランダムアクセス可能なパケット記憶手段として機能するものであり、その機能は例えばSRAM、DRAMなどにより実現される。そして、本実施形態では、図7に示すように、このRAM80がヘッダ領域（広義には制御情報領域）とデータ領域に分離されている。そして、パケットのヘッダ（広義には制御情報）は図7のヘッダ領域に格納され、パケットのデータはデータ領域に格納される。

【0068】なおRAM80は、本実施形態のデータ転送制御装置に内蔵させることが特に望ましいが、その一部又は全部を外付けにすることも可能である。

【0069】バス90（或いはバス92、94）は、アプリケーションに接続されるものである（第1のバス）。またバス96（或いはバス98）はデータ転送制御装置をコントロールするためのものであり、データ転送制御装置をコントロールするデバイス（例えばCPU）に電氣的に接続される（第2のバス）。またバス100（或いはバス102、104、105、106、107、108、109）は、物理層のデバイス（例えば

PHYチップ）に電氣的に接続されるものである（第3のバス）。また、バス110は、ランダムアクセス可能な記憶手段であるRAMに電氣的に接続されるものである（第4のバス）。

【0070】バッファマネージャ70の調停回路74は、DMAC40、DMAC42、DMAC44、CPUインターフェース60、DMAC54からのバスアクセス要求の調停を行う。そして、この調停結果に基づいて、各々、バス105、107、109、98、94のいずれかと、RAM80のバス110との間にデータの経路が確立される（第1、第2、第3のバスのいずれかと第4のバスとの間にデータ経路が確立される）。

【0071】本実施形態の1つの特徴は、ランダムアクセスが可能でありパケットを格納するRAM80を設けると共に、互いに分離されるバス90、96、100と、これらのバスとRAM80のバス110との間にデータ経路を確立するための調停回路74とを設けた点にある。

【0072】例えば図8に、本実施形態と構成の異なるデータ転送制御装置の例を示す。このデータ転送制御装置では、リンクコア902は、PHYインターフェース900、バス922を介してPHYチップと接続される。また、リンクコア902は、FIFO904、906、908、CPUインターフェース910、バス920を介してCPU912に接続される。そして、CPU912は、バス924を介して、CPUにローカルなメモリであるRAM914に接続される。

【0073】なお、FIFO904、906、908は、図6のFIFO30、32、34と異なり、非常に段数の多いものとなる（例えば1つのFIFOが16段程度）。

【0074】図8の構成のデータ転送制御装置を用いた場合のデータ転送の手法について図9を用いて説明する。PHYチップ930を介して他のノードから送られてきた受信パケットは、バス922、データ転送制御装置932、バス920を介してCPU912が受け取る。そして、CPU912は、受け取った受信パケットをバス924を介してRAM914に書き込む。そして、CPU912は、受信パケットをアプリケーション層が使用できるように加工し、バス926を介してアプリケーション層のデバイス934に転送する。

【0075】一方、アプリケーション層のデバイス934からのデータを転送する場合には、CPU912は、このデータをRAM914に書き込む。そして、RAM914のデータにヘッダを付加することでIEEE1394に準拠したパケットを生成する。そして生成されたパケットは、データ転送制御装置932、PHYチップ930などを介して他のノードに送信される。

【0076】しかしながら、このようなデータ転送手法によると、CPU912の処理負荷が非常に重くなる。

従って、ノード間を接続するシリアルバスの転送速度が高速になっても、CPU912の処理のオーバーヘッドなどに起因して、システム全体の実転送速度は低くなり、結局、高速なデータ転送を実現できない。

【0077】このような問題を解決する1つの手法として、図10に示すように、データ転送制御装置932とRAM914との間でのデータ転送や、RAM914とアプリケーション層のデバイス934との間でのデータ転送を、ハードウェアDMAにより実現する手法も考えられる。

【0078】しかしながら、この手法では、CPUバス928が、データ転送制御装置932、RAM914間でのデータ転送、RAM914、CPU912間でのデータ転送、RAM914、アプリケーション層デバイス934間でのデータ転送に使用されることになる。従って、システム全体のデータ転送の高速化を図ろうとすると、CPUバス928としてPCIバスのような高速なバスを使用しなければならなくなり、これは、データ転送制御装置を使用する電子機器の高コスト化を招く。

【0079】これに対して、本実施形態では図11に示すように、データ転送制御装置120、アプリケーション層デバイス124間のバス90と、CPUバス96と、データ転送制御装置120、RAM80間のバス110とが互いに分離されている。従って、CPUバス96をデータ転送の制御のみに使用できるようになる。また、バス90を占有して、データ転送制御装置120、アプリケーション層デバイス124間でデータ転送を行うことができるようになる。例えば、データ転送制御装置120が組み込まれる電子機器がプリンタである場合には、バス90を占有して印字データを転送できるようになる。この結果、CPU66の処理負荷を軽減でき、システム全体の実転送速度を高めることができる。またCPU66として安価なものを採用できると共に、CPUバス96として高速なバスを使用する必要性がなくなる。このため、電子機器の低コスト化、小規模化を図れるようになる。

### 【0080】3. ヘッダ領域とデータ領域の分離

#### 3.1 本実施形態の特徴

図8の比較例では、図12(A)に示すように、リンクコア902とCPU912の間には、送信用のFIFO904、906と受信用のFIFO908が設けられている。

【0081】即ち、他のノードから送られてくる受信パケットは、リンクコア902を介してFIFO908に順次入力される。そして、CPU912(CPU上で動作するファームウェアやアプリケーションソフト)は、FIFO908に入力された順序で受信パケットをFIFO908から順次読み出す。

【0082】また、CPU912は、転送すべき送信パケットをFIFO904、906に順次入力する。そし

て、送信パケットは、このFIFO904、906に入力された順序でリンクコア902を介して他のノードに順次転送される。

【0083】これに対して、本実施形態では、図12

(B)に示すように、ノード間でのパケット転送のための各種サービスを提供するリンクコア20と、CPU66との間に、FIFO30、32、34の他に、ランダムアクセス可能なパケット記憶手段として機能するRAM80を介在させている。

10 【0084】即ち、他のノードからリンクコア20を介して送られてくる受信パケットは、FIFO34に格納された後に、DMAC44(書き込み手段)によりRAM80に書き込まれる。そして、CPU66が、このランダムアクセス可能なRAM80から受信パケットを読み出す。

【0085】また、CPU66などにより、転送すべき送信パケットがRAM80に書き込まれる。そして、この書き込まれた送信パケットは、DMAC40、42

(読み出し手段)により読み出され、FIFO30、32、リンクコア20を介して他のノードに転送される。

20 【0086】図12(A)では、CPU912は、受信した順序でFIFO908から受信パケットを受け取らなければならない。送信する順序でFIFO904、906に送信パケットを格納しなければならない。これに対して、リンクコア20とCPU66との間にランダムアクセス可能なRAM80を介在させる本実施形態では、受信順序に依存せずに、RAM80の所望のアドレスに受信パケットを書き込むことが可能になる。また、送信順序に依存せずに、CPU66などがRAM80の所望のアドレスに送信パケットを書き込むことも可能になる。

【0087】また、図12(A)では、FIFO904、906、908の段数を非常に多くする必要がある。これに対して、本実施形態では、FIFO30、32、34の段数を格段に減らすことができ、例えば段数を3段以下とすることも可能になる。

【0088】なお、本実施形態では、FIFO30、32、34を設けない構成にしてもよい。

40 【0089】また、本実施形態では、図7に示すように、RAM80の記憶領域を、パケットのヘッダ(広義には制御情報)が格納されるヘッダ領域(広義には制御情報領域)とパケットのデータが格納されるデータ領域に分離している。これは、図12(B)に示すようにリンクコア20とCPU66との間にランダムアクセス可能なRAM80を介在させることで実現可能になる。

50 【0090】即ち、図12(A)では、CPU912は、受信した順序でFIFO908から受信パケットを受け取らなければならない。従って、受信パケットをヘッダとデータに分離する処理を実現しようとする、FIFO908から読み出した受信パケットをCPU91

2がローカルメモリであるRAMに一旦書き込み、そのRAMからCPU912が受信パケットを読み出してヘッダとデータに分離する必要がある。また、図12

(A)では、CPU912は、送信する順序でFIFO904、906に送信パケットを入力しなければならない。例えば、パケット1（ヘッダ1、データ1）、パケット2（ヘッダ2、データ2）、パケット3（ヘッダ3、データ3）を送信する場合には、ヘッダ1、データ1、ヘッダ2、データ2、ヘッダ3、データ3というような順序でFIFO904、906に送信パケットを入力しなければならない。従って、CPU912による並べ替え処理が必要になる。

【0091】このように図12（A）では、CPU912の処理負荷が非常に重くなり、これは結局、システム全体の実転送速度の低下という事態を招く。

【0092】これに対して、本実施形態では、RAM80の記憶領域がヘッダ領域とデータ領域に分離される。より具体的には、図13に示すように、各受信パケットのヘッダとデータがハードウェアにより分離され、ヘッダがヘッダ領域に、データがデータ領域に格納される。また、図14に示すように、ヘッダ領域に格納されたヘッダとデータ領域に格納されたデータとがハードウェアにより結合され、各ノードに転送すべき送信パケットが組み立てられる。従って、CPU66の処理負荷が図12（A）に比べて非常に軽くなり、システム全体の実転送速度を向上できる。また、CPU66として安価なものを採用できると共にCPU66に接続されるバスも低速なもので済むため、データ転送制御装置や電子機器の小規模化、低コスト化を図れるようになる。

【0093】また、本実施形態によれば、ヘッダはヘッダ領域にまとめられて格納され、データもデータ領域にまとめられて格納される。従って、ヘッダやデータの読み出し処理や書き込み処理の簡易化を図ることができ、処理のオーバーヘッドを軽減化できる。例えば、図11の手法でデータ転送を行う場合には、CPU66は、CPUバス96を介してヘッダ領域にのみアクセスしヘッダを読み出したり書き込んだりするだけでデータ転送を制御できるようになる。また、アプリケーション層のデバイス124は、データ領域のデータを連続してバス90を介して読み出したり、データを連続してデータ領域に書き込んだりすることが可能になる。

【0094】なお、図15に示すように、ヘッダ領域に格納される各ヘッダと、データ領域に格納される各データとは、ヘッダに含ませたデータポイントにより対応させることが望ましい。この場合、受信パケットのヘッダへのデータポイントの付加は、例えばリンクコア20が行い、送信パケットのヘッダへのデータポイントの付加は、例えばファームウェアが行うようにする。

【0095】また、本実施形態では、図16に示すように、受信パケットのスタート（ヘッダの最初）、ヘッ

ダ、データ、トレーラーを区別するためのTAGを生成し、このTAGを受信パケットに関連づけている。より具体的には、図6のリンクコア20が、受信パケットのスタート、ヘッダ、データ、トレーラーをFIFO34に転送する際に、図16に示す2ビットのTAGも同時にFIFO34に転送している。そして、本実施形態では、受信パケットに関連づけられたこのTAGを利用して、図13に示すように受信パケットをヘッダとデータに分離しヘッダ領域とデータ領域に格納している。より具体的には、図6のDMAC44が、受信パケットと共にFIFO34から出力されるTAGを利用して、受信パケットをヘッダとデータに分離し、RAM80に書き込んでいる。なお、TAGは、少なくともヘッダとデータを区別できるものであればよい。

【0096】例えば、TAGを利用せずに受信パケットのヘッダとデータを分離する手法として、ヘッダに含まれるtcode（トランザクションコード）を利用する手法が考えられる。即ち、tcodeをデコードして、ヘッダのサイズを調べ、そのサイズの分だけ受信パケットをヘッダ領域に格納し、残りをデータ領域に格納する。

【0097】しかしながら、この手法では、tcodeをデコードする回路が必要となり、回路が大規模化してしまう。

【0098】これに対して、TAGを利用すれば、DMAC44は、TAGを見るだけで受信パケットをヘッダとデータに分離できる。従って、回路をそれほど大規模化することなく、簡易な処理で受信パケットのヘッダとデータを分離できるようになる。

【0099】また、本実施形態では、図17（A）に示すように、ヘッダ領域とデータ領域の他に、これらの領域と分離されたCPU66のワーク領域をRAM80に設けている。このようにすれば、CPU66が、ワーク領域に直接アクセスして、CPU66の処理のためにその領域を利用できるようになる。従って、CPU66がローカルメモリを有する場合には、そのメモリを小容量化することが可能になる。更に、CPU66のローカルメモリを設ける必要性自体を無くすことも可能になる。これにより、データ転送制御装置や電子機器の小規模化、低コスト化を図れるようになる。

【0100】また、本実施形態では、図17（B）に示すように、ヘッダ領域を受信用と送信用の領域に分離（分割）している。これらの受信用、送信用のヘッダ領域の各々は、少なくとも1個以上のパケットを格納できるものであればよい。

【0101】例えば図8の構成では、ローカルメモリであるRAM914にCPU912が受信パケットや送信パケットを格納する。このため、CPU912は、受信パケットや送信パケットを格納したアドレスを各パケット毎に保持し管理しなければならない。従って、CPU

912の処理が煩雑化し、処理負担が増加する。特に、CD-Rドライブのように受信処理と送信処理が混在するアプリケーションでは、受信パケットと送信パケットが混在してRAM914に格納されることになるため、処理の煩雑化、処理負担の増加の問題は更に深刻になる。

【0102】これに対して、本実施形態によれば、ヘッダ領域が受信用と送信用の領域に分離される。従って、RAM80の受信用ヘッダ領域には受信用のヘッダが、送信用ヘッダ領域には送信用のヘッダが規則的に格納される。従って、CPU66は、受信処理の際には受信用ヘッダ領域にアクセスし、送信処理の際には送信用ヘッダ領域にアクセスすればよいことになる。そして、例えば受信パケットを連続して受信する場合には、受信用ヘッダ領域からヘッダを連続して読み出し、送信パケットを連続して送信する場合には、送信用ヘッダ領域にヘッダを連続して書き込めばよいことになる。従って、CPU66の処理負担を大幅に軽減でき、システム全体の実転送速度の向上を期待できるようになる。

【0103】また、本実施形態では、図17(C)に示すように、データ領域を受信用と送信用の領域に分離している。

【0104】このようにすれば、例えば図11において、アプリケーション層のデバイス124は、RAM80の受信用データ領域から連続して受信データを読み出せるようになると共に、送信用データ領域に連続して送信データを書き込めるようになる。そして、読み出しや書き込みの際にはアドレスを単純にインクリメント（或いはデクリメント）すればよいので、アドレス制御も容易になる。この結果、本実施形態によれば、処理のオーバーヘッドを格段に軽減でき、実転送速度の向上を期待できるようになる。

【0105】また、本実施形態では、データ領域を、アイソクロナス転送用と非同期転送用の領域に分離している。より具体的には例えば図17(D)に示すように、データ領域を、アイソクロナス受信用、非同期受信用、アイソクロナス送信用、非同期送信用の領域に分離している。

【0106】例えば図8の構成では、アイソクロナスパケットと非同期パケットとが混在してFIFO908に入力される。そして、CPU912は、この入力された順序でFIFO908からパケットを読み出す。また、この読み出した順序でパケットをローカルメモリであるRAM914に格納する。従って、RAM914には、アイソクロナスパケットと非同期パケットが混在して格納されることになる。

【0107】さて、アイソクロナスパケットの処理にはリアルタイム性が要求される。従って、CPU912は一定期間内にアイソクロナスパケットに関する処理を終了しなければならない。ところが、図8では、CPU9

12はFIFO908に入力された順序でFIFO908からパケットを読み出さなければならない、また、RAM914には、アイソクロナスパケットと非同期パケットが混在するようになる。従って、例えば非同期パケットの次にアイソクロナスパケットが送られてきた場合にも、非同期パケットを先に処理しなければならない、アイソクロナスパケットを先に処理することができない。このため、リアルタイム性が要求されるアイソクロナスパケットの処理が間に合わなくなる可能性がある。

【0108】これに対して、本実施形態では、アイソクロナスパケットはアイソクロナス転送用の領域に格納され、非同期パケットは非同期転送用の領域に格納される。従って、非同期パケットの次にアイソクロナスパケットが送られてきた場合にも、アイソクロナスパケットを先に読み出し、先に処理することが可能になる。この結果、アイソクロナスパケットの処理を一定期間内に完了でき、アイソクロナスパケットの処理に要求されるリアルタイム性を維持できるようになる。

【0109】また、データ領域をアイソクロナス転送用と非同期転送用の領域に分離することで、アプリケーション層のデバイスの処理も簡易化できる。例えばデジタルビデオカメラにおいて、リアルタイム性が要求される動画像をアイソクロナスで転送し、信頼性が要求される静止画像を非同期で転送する場合を考える。そして、図11において、データ転送制御装置120とアプリケーション層のデバイス124との間でバス90を用いて直接に動画像や静止画像のデータを転送したとする。この場合に、データ領域をアイソクロナス転送用と非同期転送用の領域に分離することにより、バス90を用いた動画像や静止画像のデータの転送を簡易化できる。動画像のデータを転送する場合には、動画像のデータをまとめてアイソクロナス転送用のデータ領域に書き込み、静止画像のデータを転送する場合には、静止画像のデータをまとめて非同期転送用のデータ領域に書き込めばよいからである。また、この書き込みの際には、アドレスを単純にインクリメント（或いはデクリメント）すればよいので、アドレス制御も容易になるからである。

【0110】また、本実施形態では、RAM80が複数の領域に分離される場合において、各領域のサイズを可変に制御するようにしている。より具体的には図18に示すように、各領域の境界のアドレスを指すポインタP1～P6を可変に制御する。

【0111】このようにすれば、アプリケーションに応じた最適な領域分割を実現できる。例えば、プリンタのように、パケットを送信することが少なく、パケットを受信することが多いアプリケーションでは、受信用の領域のサイズを大きくする。また、スキャナのように、パケットを受信することが少なく、パケットを送信することが多いアプリケーションでは、送信用の領域のサイズを大きくする。また、非同期転送を行うことが多いアプ

リケーションでは非同期転送用の領域のサイズを大きくし、アイソクロナス転送を行うことが多いアプリケーションではアイソクロナス転送用の領域のサイズを大きくする。或いは、非同期転送のみを行うアプリケーションでは、アイソクロナス転送用の領域を零にする。このようにすることにより、限られたリソース（少ない記憶容量のRAM80）を有効利用することが可能になる。

【0112】また、図8の構成でFIFO904、906、908の各々のサイズを可変にする手法には、ハードウェアの構成が複雑になるという問題点がある。これに対して、RAM80の各領域のサイズを可変制御するためには、ポインタP1～P6の指すアドレスを制御する（ポインタの指すアドレスが格納されるレジスタの内容を制御する）のみでよい。このため、簡易なハードウェア構成で各領域のサイズを可変に制御できる。

【0113】なお、RAM80の各領域のサイズは、電源投入後もダイナミックに可変制御できることが望ましい。例えば、CD-Rドライブのようなアプリケーションでは、CD-Rへのデータの書き込み時には受信処理が主に行われ、CD-Rからのデータの読み出し時には送信処理が主に行われる。従って、CD-Rへのデータの書き込み時には、受信用の領域のサイズが大きくなるようにポインタをダイナミックに切り替える。また、CD-Rからのデータの読み出し時には、送信用の領域のサイズが大きくなるようにポインタをダイナミックに切り替える。このようにすることで、限られたリソースを有効利用することが可能になる。

#### 【0114】3. 2 受信側の構成

次に受信側の構成について説明する。図19に、リンクコア20、FIFO34、DMAC44の詳細な構成の一例を示す。

【0115】リンクコア20は、バス監視回路130、直列・並列変換回路132、パケット整形回路160を含む。そして、パケット整形回路160は、パケット診断回路142、シーケンサ167、バッファ168、セクタ170を含み、パケット診断回路142は、TAG生成回路162、ヘッダ&トレイラー生成回路164、エラーチェック回路166を含む。

【0116】ここで、バス監視回路130は、PHYインターフェース10を介してPHYチップに接続される8ビット幅のデータバスD、2ビット幅のコントロールバスCTLを監視する回路である。

【0117】直列・並列変換回路132は、データバスDのデータを32ビットのデータに変換する回路である。例えば、転送速度が400Mbpsの場合には8ビットのデータが32ビットのデータに、200Mbpsの場合には4ビットのデータが32ビットのデータに、100Mbpsの場合には2ビットのデータが32ビットのデータに変換される。

【0118】パケット診断回路142は、パケットの診

断を行う回路である。TAG生成回路162は、ヘッダ、データ、トレイラーなどを区別するためのTAGを生成する回路であり、ヘッダ&トレイラー生成回路164は、ヘッダ及びトレイラー（フッター）を生成する回路である。また、エラーチェック回路166は、パケットに含まれるパリティなどのエラーチェック情報をチェックしてエラーを検出する回路である。

【0119】シーケンサ167は各種の制御信号を生成するものである。バッファ168、セクタ170は、直列・並列変換回路132からのDI、パケット診断回路142からのヘッダ及びトレイラー、DMAC44からのデータポインタのいずれかを、パケット診断回路142からの信号SELにより選択するためのものである。

【0120】FIFO34は、リンクコア20からの出力データであるRDの位相と、RAM80への書き込みデータであるWDATAの位相とを調整するためのバッファとして機能するものであり、FIFO状態判断回路35を含む。FIFO状態判断回路35は、FIFOが空になると、EMPTYをアクティブにし、FIFOがフルになると、FULLをアクティブにする。

【0121】DMAC44は、パケット分離回路180、アクセス要求実行回路190、アクセス要求発生回路192を含む。

【0122】パケット分離回路180は、パケット整形回路160により整形されたパケットを分離して、ヘッダ及びトレイラーをRAM80のヘッダ領域に、データをデータ領域に書き込むための回路である（図13参照）。パケット分離回路180は、TAG判別回路182、ポインタ更新回路184、アドレス発生回路188を含む。

【0123】TAG判別回路182は、TAG生成回路162により生成されたTAG（DTAG）を判別する回路である。

【0124】ポインタ更新回路184は、TAG判別回路182の出力を受け、RAM80にヘッダやデータを書き込むためのヘッダポインタやデータポインタを更新するための回路である。

【0125】アドレス発生回路188は、ポインタ更新回路184の出力を受け、RAM80への書き込みアドレスWADRを発生する回路である。

【0126】アクセス要求実行回路190は、リンクコア20からのアクセス要求を実行するための回路である。アクセス要求実行回路190は、FIFO状態判断回路35からのFULLがアクティブになると、FULLをアクティブにする。パケット整形回路160内のシーケンサ167は、FULLがアクティブでないことを条件に、RD（RxData）のストロブ信号であるRDSをアクティブにする。

【0127】なおRFAILは、受信における失敗を、

シーケンサ167がアクセス要求実行回路190に対して知らせるための信号である。

【0128】アクセス要求発生回路192は、RAM80へのアクセス要求を発生するための回路である。アクセス要求発生回路192は、バッファマネージャ70からの書き込みアクノリッジメントであるWACKやFIFO状態判断回路35からのEMPTYを受け、書き込み要求であるWREQをバッファマネージャ70に出力する。

### 【0129】3.3 受信側の動作

次に、受信側の動作の詳細について図20のタイミング波形図などを用いて説明する。

【0130】まず、リンクコア20の動作について説明する。

【0131】PHYチップを介して他のノードからのパケットを受信すると、パケット診断回路142がそのパケットを診断する。そして、ヘッダ&トレイラー生成回路164がヘッダを生成(整形)する。このヘッダは、バッファ168を介してセクタ170に入力され、パケット診断回路142からの信号SELに基づきセクタ170がこのヘッダを選択する。これにより、図20のA1に示すように、RDとしてヘッダ(H0~H4)がFIFO34に出力されることになる。

【0132】なお、図21(A)に、シリアルバス上で転送される非同期パケットのフォーマット(IEEE1394規格)を示す。一方、図21(B)に、RAM80のヘッダ領域に格納される非同期受信パケットのヘッダ部分のフォーマットを示す(同図において網掛けとなっている部分がトレイラーである)。このように本実施形態では、図21(A)に示すフォーマットのパケットを、ファームウェアなどの上層が使用できるように、図21(B)に示すフォーマットの packets に整形している。

【0133】また本実施形態では、ヘッダの第4クワドレットであるH4(図20のA2)は、図21(B)に示すようにデータ領域からデータを取り出すためのデータポインタになっている。このデータポインタ(H4)は、DMAC44(ポインタ更新回路184)からバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。このように、パケット整形回路160は、DMAC44からデータポインタを受け取り、そのデータポインタを、RAM80に書き込まれるヘッダに埋め込んでいる。

【0134】次に、PHYチップから、データバスDを介してパケットのデータ部分が送られてくる。直列・並列変換回路132は、このデータ部分を32ビットのデータであるDIに変換し、パケット診断回路142及びバッファ168に出力する。

【0135】なお、DIEは、DIが有効か無効かを示す信号であり、DISは、DIの取り込みタイミングを

知らせるためのストロブ信号である。

【0136】直列・並列変換回路132からのDIは、バッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、A3に示すように、RDとしてデータD0~DnがFIFO34に出力される。

【0137】次に、ヘッダ&トレイラー生成回路164からのトレイラーがバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、A4に示すように、RDとしてトレイラー(H5.図21(B)の網掛け部分)がFIFO34に出力される。

【0138】さて、TAG生成回路162は、RDとして出力される情報を区別するためのTAGを生成している。本実施形態では図16に示すようにTAGは2ビットであり、(00)、(01)、(10)、(11)は、各々、ヘッダ、トレイラー、データ、スタート(ヘッダの最初)を表す。従って、例えば図20では、(11)、(00)、…、(10)、…、(01)というようにTAGが変化する。FIFO34には、この2ビットのTAGと32ビットのRDとからなる34ビットのデータが入力されることになる。

【0139】次に、FIFO34の動作について説明する。

【0140】FIFO34は、リンクコア20からのTAG、RDを受け、A5、A6に示すように、DTAG、WDATAとして出力する。

【0141】FIFO34内のFIFO状態判断回路35は、内蔵するカウンタにより、FIFO34のデータ数(FIFOカウント)をカウントする。そして、FIFO34がフル(データ数=2)になった場合には、図20のA7に示すようにFULLをアクティブ(Hレベル)にする。また、FIFO34が空(データ数=0)になった場合には、A8に示すようにEMPTYをアクティブにする。FIFO34がフルになったことは、FULL、FFULLにより、DMAC44内のアクセス要求実行回路190やリンクコア20内のシーケンサ167に伝えられる。また、FIFO34が空になったことは、EMPTYにより、DMAC44内のアクセス要求発生回路192に伝えられる。

【0142】次に、DMAC44の動作について説明する。

【0143】アクセス要求発生回路192は、A9に示すようにEMPTYが非アクティブ(Lレベル)になったこと(FIFO34が空でないこと)を条件に、A10に示すようにWREQをアクティブにする。そして、バッファマネージャ70からWACKを受け取ると、WREQを非アクティブにする。

【0144】さて、本実施形態では、受信時のバス調停においては、DMAC44からのアクセス要求の優先順

位を最も高くしている。従って、A10、A11に示すように、DMAC44からのWREQと、CPUインターフェース60やポート用のDMAC54からのOtherWREQとが競合した場合には、WREQの方が優先する。即ち、A12、A13に示すように、WACKの方がOtherWACKよりも先にアクティブになる。このように、WREQとOtherWREQが競合した場合に、WREQの方を優先させるのは、以下の理由による。即ち、IEEE1394では、SCSIなどとは異なり、他のノードからのパケットが転送クロックに同期して次々に転送されてくる。従って、これらの間断なく転送されてくるパケットを優先してRAM80に次々に格納する必要があるからである。

【0145】また、本実施形態では、CPUインターフェース60やポート用のDMAC54からのアクセス要求をバッファマネージャ70が受け付けている間は、A14に示すように、DMAC44のアクセス要求は所与の期間だけ待たされる。従って、リンクコア20からのRDと、バッファマネージャ70へのWDATAとは同期しない。この理由により、本実施形態では、RDとWDATAの位相を調整するためのFIFO34を設けている。この場合、FIFO34は、位相調整のために必要な最低限の段数（好ましくは3段以下、更に好ましくは2段以下）を備えていればよい。

【0146】パケット分離回路180に含まれるTAG判別回路182は、WDATAと共にFIFO34から出力されるDTAGを判別し、WDATAが、スタート（ヘッダの最初）、ヘッダ、データ、トレイラーのいずれなのかを判定する。そして、ポイント更新回路184は、この判定結果に基づいて、ヘッダポイントやデータポイントの更新を行う。次に、アドレス発生回路188は、更新されたヘッダポイントやデータポイントに基づいて、WDATAの書き込みアドレスであるWADRを発生する。

【0147】より具体的には、例えば、WDATAがスタート又はヘッダであるとDTAGに基づき判定された場合は、ポイント更新回路184が、図22(A)に示すように、ヘッダポイントHPのインクリメント（広義には更新）を行う。そして、アドレス発生回路188は、図20のA15に示すように、インクリメントされるヘッダポイントに応じたWADRを発生する。

【0148】次に、WDATAがデータであるとDTAGに基づき判定された場合は、ポイント更新回路184が、図22(B)に示すように、データポイントDPのインクリメントを行う。このデータポイントDPは、パケット整形回路160によりヘッダの第4クワドレットに埋め込まれたH4に相当する。アドレス発生回路188は、図20のA16に示すように、インクリメントされるデータポイントに応じたWADRを発生する。

【0149】次に、WDATAがトレイラーであるとD

TAGに基づき判定された場合は、ポイント更新回路184が、図22(C)に示すように、ヘッダポイントのインクリメントを行う。そして、アドレス発生回路188は、図20のA17に示すように、インクリメントされるヘッダポイントに応じたWADRを発生する。

【0150】なお、最終的には図22(D)に示すように、ヘッダポイントは、処理対象となったパケットのヘッダ部分の下側の境界（次のパケットのヘッダ部分の上側の境界）を指すようになる。また、データポイントは、パケットのデータ部分の下側の境界（次のパケットのデータ部分の上側の境界）を指すようになる。ヘッダポイントやデータポイントの最終位置は、受信に失敗がなかった（RFAILが非アクティブ）ことを条件に、図6のレジスタ46内のヘッダポイント設定レジスタやデータポイント設定レジスタにリストアされる。

【0151】以上のようにして、パケットを分離してヘッダ領域とデータ領域に書き込むことが可能になる。

【0152】特に本実施形態では、ヘッダに付加されるデータポイントが、ポイント更新回路184からパケット整形回路160に伝えられる。そしてパケット整形回路160が、この伝えられたデータポイントをパケットのヘッダに付加する。このようにすることで、ヘッダ領域からヘッダを読み出したファームウェアなどが、そのヘッダに対応するデータのデータ領域での格納アドレスを容易に知ることができるようになる。また、データポイントの付加は、パケット整形回路160により行われ、DMAC44はこれに関与する必要がない。従って、DMAC44がRAM80へのデータ書き込み処理に専念できるようになり、DMAC44の回路構成や処理を簡素化できるようになる。

【0153】なお、ヘッダ領域とデータ領域の境界などの、RAM80を分離する領域の境界（図18のP1～P6）の設定は、CPUインターフェース60を介してCPU66（ファームウェア等）が、図6のレジスタ46に含まれるポイント設定レジスタに対して、境界のアドレスを指すポイントを設定することで実現される。

【0154】また、データ領域が複数の領域に分離される場合（例えば図18に示すようにアイソクロナス転送用と非同期転送用の領域に分離される場合、或いは第1、第2の非同期転送用の領域に分離される場合等）には、例えばtcodeなどのパケットの制御情報に基づき、分離された複数の領域のいずれかにパケットのデータを書き込むようにすることが望ましい。

【0155】より具体的には、図19に示すように、DMAC44が、複数のデータポイント、例えば第1、第2のデータポイントをパケット整形回路160に対して渡すようにする（3個以上のデータポイントを渡してもよい）。そして、パケット整形回路160は、例えば、アイソクロナス転送時（或いは第2の非同期転送時）にはDMAC44からの第1のデータポイントを選択し、

10

20

30

40

50



非同転送時（或いは第1の非同転送時）にはDMAC 44からの第2のデータポインタを選択するようにする。即ち、パケット整形回路160内のパケット診断回路142が、*t c o d e*などのパケットの制御情報に基づいて、アイソクロナス転送か非同転送か（或いは第2の非同転送か第1の非同転送か）を判別し、この判別結果に基づき信号SELを制御する。そして、バッファ168を介してセクタ170に入力される第1、第2のデータポインタのいずれかを選択するようにする。これにより、アイソクロナス転送（或いは第2の非同転送）のパケットには第1のデータポインタが埋め込まれ、非同転送（或いは第1の非同転送）のパケットには第2のデータポインタが埋め込まれるようになる。この結果、データ領域を分離する特定の領域に、連続的にデータを格納することが可能になる。即ち、デジタルカメラにおける動画データやアイソクロナス転送用データ領域に連続的に格納したり、プリンタにおける印字データを第2の非同転送用データ領域に連続的に格納したりすること（第1の非同転送用データ領域にはコマンドデータ、ステータスデータなどの制御用データを格納する）が可能になる。

#### 【0156】3. 4 送信側の構成

次に送信側の構成について説明する。図23に、FIFO30、DMAC40の詳細な構成の一例を示す。

【0157】位相調整用のバッファとして機能するFIFO30は、FIFO状態判断回路31を含む。FIFO状態判断回路31は、FIFOが空になると、EMPTYをアクティブにし、FIFOがフルになると、FULLをアクティブにする。

【0158】DMAC40は、パケット結合回路280、アクセス要求実行回路290、アクセス要求発生回路292、ACK書き込み要求発生回路294、ACK書き込みデータ&アドレス発生回路296を含む。

【0159】パケット結合回路280は、ヘッダをRAM80のヘッダ領域から読み出し、データをデータ領域から読み出し、これらのヘッダとデータによりフレームが構成される送信パケットを組み立てる回路である（図14参照）。パケット結合回路280は、ポインタ更新回路284、アドレス発生回路288を含む。

【0160】ポインタ更新回路284は、RAM80からヘッダやデータを読み出すためのヘッダポインタやデータポインタを更新するための回路であり、データポインタ取得回路285を含む。データポインタ取得回路285は、RAM80から読み出されたRDATAからデータポインタを取得する回路であり、*t c o d e*判別回路286を含む。

【0161】アドレス発生回路288は、ポインタ更新回路284の出力などを受け、RAM80の読み出しアドレスRADRを発生する回路である。

【0162】アクセス要求実行回路290は、FIFO

状態判断回路31からのEMPTYがアクティブになると、FIFOINをアクティブにする。リンクコア20は、FIFOINがアクティブでないことを条件に、TD(TxDat a)のストローブ信号であるTDSをアクティブにする。

【0163】なおTFAILは、送信における失敗を、リンクコア20がアクセス要求実行回路290に対して知らせるための信号である。

【0164】アクセス要求発生回路292は、バッファマネージャ70からの読み出しアクノリッジメントであるRACKやFIFO状態判断回路31からのFULLを受け、読み出し要求であるRREQをバッファマネージャ70に出力する。

【0165】ACK書き込み要求発生回路294は、リンクコア20からのTCMPやバッファマネージャ70からのWACKを受け、WREQをバッファマネージャ70に出力する。また、ACK書き込みデータ&アドレス発生回路296は、リンクコア20からTACKを受け、送信パケットに書き戻すACKのコードをWDATAとして出力し、ACKを書き戻すアドレスをWADRとして出力する。

#### 【0166】3. 5 送信側の動作

次に、送信側の動作の詳細について図24のタイミング波形図などを用いて説明する。

【0167】まず、リンクコア20の動作について説明する。

【0168】送信開始を知らせるTSTARTがアクティブになると、図24のB1に示すように、リンクコア20は、ストローブ信号であるTDSを用いてFIFO30からTDを取り込む。この場合、リンクコア20には、ヘッダ(H0~H3)、データ(D0~Dn)の順でTDSが取り込まれる。

【0169】なお、図25に、RAM80のヘッダ領域に格納される非同転送送信パケットのヘッダ部分のフォーマットを示す。同図に示すようにヘッダの第4クワドレットはデータポインタになっている。

【0170】さて、図24のB2に示す位置では、リンクコア20は、TDSをアクティブにしない。従って、B3に示すように、ヘッダの第4クワドレットであるH4はリンクコア20に取り込まれない。図25に示すように第4クワドレットのH4はデータポインタであり、リンクコア20は、このデータポインタを必要としないからである。そして、リンクコア20は、B3に示す期間において、ヘッダCRC（図21(A)参照）を生成しヘッダに付加する処理を行っている。

【0171】1つのパケットの送信処理が終了すると、リンクコア20は、B4に示すようにTCMPをアクティブにする。そして、送信のデスティネーションノードからPHYチップを介して返されてきたACKのコードを（図1(A)参照）、B5に示すようにTACKとし

10

20

30

40

50



てDMAC40に出力する。このACKのコードは、ACK書き込み要求発生回路294、ACK書き込みデータ&アドレス発生回路296により、RAM80のヘッダ領域に格納されているヘッダに書き戻されることになる(図25の第7クワドレット)。

【0172】次に、FIFO30の動作について説明する。

【0173】FIFO30は、バッファマネージャ70からのRDATAを受け、TDとしてリンクコア20に出力する。

【0174】FIFO30内のFIFO状態判断回路31は、内蔵するカウンタにより、FIFO30のデータ数(FIFOカウント)をカウントする。そして、FIFO30が空(データ数=0)になった場合には、図24のB6に示すようにEMPTYをアクティブにする。また、FIFO30がフル(データ数=2)になった場合には、B7に示すようにFULLをアクティブ(Hレベル)にする。FIFO30が空になったことは、EMPTY、FIFOINにより、DMAC40内のアクセス要求実行回路290やリンクコア20に伝えられる。また、FIFO30がフルになったことは、FULLにより、DMAC40内のアクセス要求発生回路292に伝えられる。

【0175】次に、DMAC40の動作について説明する。

【0176】アクセス要求発生回路292は、B8に示すように、FULLが非アクティブ(Lレベル)であること(FIFO34がフルでないこと)を条件にRREQをアクティブにする。そして、バッファマネージャ70からRACKを受け取ると、RREQを非アクティブにする。

【0177】なお、本実施形態では、送信時のバス調停においては、DMAC40(或いはDMAC42)からのアクセス要求の優先順位を最も高くしている。従って、DMAC40からのRREQと、CPUインターフェース60やポート用のDMAC54からのアクセス要求(OtherRREQ)とが競合した場合には、RREQの方が優先する。一方、RREQよりも先に、CPUインターフェース60やポート用のDMAC54からのアクセス要求があった場合には、B9に示すように、DMAC40のアクセス要求は所与の期間だけ待たされる。従って、バッファマネージャ70からのRDATAと、リンクコア20へのTDとは同期しない。この理由により、本実施形態では、RDATAとTDの位相を調整するためのFIFO30を設けている。

【0178】さて、送信が開始すると、ポインタ更新回路284が、ヘッダポインタHPのインクリメント(広義には更新)を行う。そして、アドレス発生回路288は、図24のB10に示すように、インクリメントされるヘッダポインタに応じたRADRを発生する。このよ

うにして、RDATAのヘッダ部分がRAM80から順次読み出される。

【0179】RDATAとしてH4が読み出されると、パケット結合回路280に含まれるデータポインタ取得回路285が、このH4を、データポインタDPとして取得する。より具体的には、RDATAとしてH0が読み出されると、データポインタ取得回路285内のtc code判別回路286が、H0に含まれるtc code(図25参照)を判別する。そして、ヘッダの例えば第4クワドレットにデータポインタがあるとtc code(パケットフォーマット識別情報)に基づき判断された場合には、RDATAとしてH4が読み出された時に、データポインタ取得回路285がこのH4を取得する。即ち、図24のB11に示すように、RDATAのH4が、データポインタとして取得され、RADRとして出力される。

【0180】なお、本実施形態では、B3、B11に示すように、リンクコア20がヘッダCRCを生成している期間を利用して、RDATAからデータポインタであるH4を取得している。即ち、本実施形態では、ヘッダCRCの生成はリンクコア20が行い、DMAC40はこれに関与しない。一方、データポインタの取得はDMAC40が行い、リンクコア20はこれに関与しない。本実施形態では、このことに着目し、図21(A)においてヘッダCRCが配置される第4クワドレットに、図25に示すようにデータポインタを配置している。そして、ヘッダCRCが生成される期間を利用して、RDATAからデータポインタであるH4を取得するようにしている。このようにすることで、処理時間に無駄が生じるのを防止できるようになる。

【0181】データポインタが取得されると、ポインタ更新回路284が、取得されたデータポインタであるH4のインクリメントを行う。そして、アドレス発生回路288は、図24のB12に示すように、インクリメントされるデータポインタに応じたRADRを発生する。このようにして、RDATAのデータ部分がRAM80から順次読み出される。

【0182】1つのパケットの送信処理が終了し、B4に示すようにTCMPがアクティブになると、B13に示すようにACK書き込み要求発生回路294がWREQをアクティブにする。そして、リンクコア20からACK書き込みデータ&アドレス発生回路296に対してTACKを用いて送られたACKのコードが、B14に示すように、WDATAとして出力される。また、この際に、ACKのコードの書き込みアドレスであるHP+7が、WADRとして出力される。ここで、WADRがHP+7になるのは、図25に示すようにACKのコードはヘッダの第7クワドレットに書き戻されるからである。

【0183】以上のようにして、ヘッダ領域のヘッダと

データ領域のデータを結合して、送信パケットを組み立てることが可能になる。

【0184】特に本実施形態では、ヘッダとデータの結合はDMAC40が行い、リンク20はこれに関与する必要がない。従って、リンクコア20の回路構成や処理を簡素化できるようになる。

【0185】また、本実施形態では、データポインタ取得回路285が、RDATAからデータポインタ(H4)を取得し、この取得されたデータポインタに基づきRADRが発生し、データが読み出される。このようにすることで、ヘッダと、そのヘッダに対応するデータとを、確実に結合できるようになる。また、ヘッダとデータの結合処理に必要な回路構成を簡素化できるようになる。

#### 【0186】4. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器の例について説明する。

【0187】例えば図26(A)に電子機器の1つであるプリンタの内部ブロック図を示し、図27(A)にその外観図を示す。CPU(マイクロコンピュータ)510はシステム全体の制御などを行う。操作部511はプリンタをユーザが操作するためのものである。ROM516には、制御プログラム、フォントなどが格納され、RAM518はCPU510のワーク領域として機能する。表示パネル519はプリンタの動作状態をユーザに知らせるためのものである。

【0188】PHYチップ502、データ転送制御装置500を介して、パーソナルコンピュータなどの他のノードから送られてきた印字データは、バス504を介して印字処理部512に直接送られる。そして、印字データは、印字処理部512にて所与の処理が施され、プリントヘッドなどからなる印字部(データを出力するための装置)514により紙に印字されて出力される。

【0189】図26(B)に電子機器の1つであるスキャナの内部ブロック図を示し、図27(B)にその外観図を示す。CPU520はシステム全体の制御などを行う。操作部521はスキャナをユーザが操作するためのものである。ROM526には制御プログラムなどが格納され、RAM528はCPU520のワーク領域として機能する。

【0190】光源、光電変換器などからなる画像読み取り部(データを取り込むための装置)522により原稿の画像が読み取られ、読み取られた画像のデータは画像処理部524により処理される。そして、処理後の画像データがバス505を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、この画像データにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0191】図26(C)に電子機器の1つであるCD

-Rドライブの内部ブロック図を示し、図27(C)にその外観図を示す。CPU530はシステム全体の制御などを行う。操作部531はCD-Rをユーザが操作するためのものである。ROM536には制御プログラムなどが格納され、RAM538はCPU530のワーク領域として機能する。

【0192】レーザ、モータ、光学系などからなる読み取り&書き込み部(データを取り込むための装置又はデータを記憶するための装置)533によりCD-R532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、バス506を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、このデータにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0193】一方、PHYチップ502、データ転送制御装置500を介して、他のノードから送られてきたデータは、バス506を介して信号処理部534に直接送られる。そして、信号処理部534によりこのデータに所与の信号処理が施され、読み取り&書き込み部533によりCD-R532に記憶される。

【0194】なお、図26(A)、(B)、(C)において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

【0195】本実施形態のデータ転送制御装置を電子機器に用いることで、高速なデータ転送が可能になる。従って、ユーザがパーソナルコンピュータなどによりプリントアウトの指示を行った場合に、少ないタイムラグで印字が完了するようになる。また、スキャナへの画像取り込みの指示の後に、少ないタイムラグで読み取り画像をユーザは見るができるようになる。また、CD-Rからのデータの読み取りや、CD-Rへのデータの書き込みを高速に行うことができるようになる。更に、例えば1つのホストシステムに複数の電子機器を接続して利用したり、複数のホストシステムに複数の電子機器を接続して利用したりすることも容易になる。

【0196】また本実施形態のデータ転送制御装置を電子機器に用いることで、CPU上で動作するファームウェアの処理負荷が軽減され、安価なCPUや低速のバスを用いることが可能になる。更に、データ転送制御装置の低コスト化、小規模化を図れるため、電子機器の低コスト化、小規模化も図れるようになる。

【0197】なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例えば、種々の光ディスクドライブ(CDROM、DVD)、光磁気ディスクドライブ(MO)、ハードディスクドライブ、TV、VTR、ビデオカメラ、オーディオ機器、電話

機、プロジェクタ、パーソナルコンピュータ、電子手帳、ワードプロセッサなど種々のものを考えることができる。

【0198】なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0199】例えば、本発明のデータ転送制御装置の構成は、図6に示す構成が特に望ましいが、これに限定されるものではない。

【0200】また本発明では、記憶手段を複数の領域に分離することが特に望ましいが、これを分離させない構成とすることも可能である。

【0201】また、本発明は、IEEE1394規格でのデータ転送に適用されることが特に望ましいが、これに限定されるものではない。例えばIEEE1394と同様の思想に基づく規格やIEEE1394を発展させた規格におけるデータ転送にも本発明は適用できる。

【0202】

【図面の簡単な説明】

【図1】図1(A)、(B)、(C)は、非同期転送とアイソクロナス転送について説明するための図である。

【図2】図2(A)、(B)は、ツリー識別について説明するための図である。

【図3】自己識別について説明するための図である。

【図4】図4(A)、(B)、(C)、(D)は、セルフIDパケットなどの物理層のパケットのフォーマットを示す図である。

【図5】IEEE1394のプロトコル構成について示す図である。

【図6】本実施形態のデータ転送制御装置の構成例を示す図である。

【図7】ヘッダ(制御情報)領域とデータ領域の分離について説明するための図である。

【図8】本実施形態の比較例の構成例について示す図である。

【図9】図8の構成によるデータ転送の手法について説明するための図である。

【図10】データ転送の手法の他の例について説明するための図である。

【図11】本実施形態のデータ転送の手法について説明するための図である。

【図12】図12(A)、(B)は、リンクコアとCPUとの間にランダムアクセス可能なRAMを介在させる手法について説明するための図である。

【図13】受信パケットをヘッダとデータに分離してRAMのヘッダ領域とデータ領域に格納する手法について説明するための図である。

【図14】ヘッダ領域に格納されるヘッダとデータ領域に格納されるデータを結合し、送信パケットを組み立てる手法について説明するための図である。

【図15】ヘッダ領域に格納するヘッダにデータポイン

タを含ませる手法について説明するための図である。

【図16】TAGについて説明するための図である。

【図17】図17(A)、(B)、(C)、(D)は、RAMを各種の領域に分離する手法について説明するための図である。

【図18】RAMの各領域のサイズを可変に制御する手法について説明するための図である。

【図19】受信側の構成の一例を示す図である。

【図20】受信側の動作について説明するためのタイミング波形図である。

【図21】図21(A)は、IEEE1394規格の非同期パケットのフォーマットであり、図21(B)は、RAMのヘッダ領域に格納される非同期受信パケットのヘッダ部分のフォーマットである。

【図22】図22(A)、(B)、(C)、(D)は、ヘッダポインタとデータポインタの更新について説明するための図である。

【図23】送信側の構成の一例を示す図である。

【図24】送信側の動作について説明するためのタイミング波形図である。

【図25】RAMのヘッダ領域に格納される非同期送信パケットのヘッダ部分のフォーマットである。

【図26】図26(A)、(B)、(C)は、種々の電子機器の内部ブロック図の例である。

【図27】図27(A)、(B)、(C)は、種々の電子機器の外観図の例である。

【符号の説明】

10 PHYインターフェース

20 リンクコア

22 レジスタ

30 FIFO(ATF)

32 FIFO(ITF)

34 FIFO(RF)

40 DMAC(ATF用)

42 DMAC(ITF用)

44 DMAC(RF用)

50 ポートインターフェース

52 FIFO(PF)

54 DMAC(PF用)

56 レジスタ

60 CPUインターフェース

62 アドレスデコーダ

63 データ同期化回路

64 割り込みコントローラ

66 CPU

68 クロック制御回路

70 バッファマネージャ

72 レジスタ

74 調停回路

50 76 シーケンサ

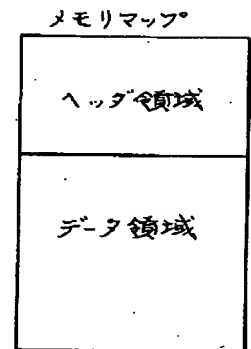
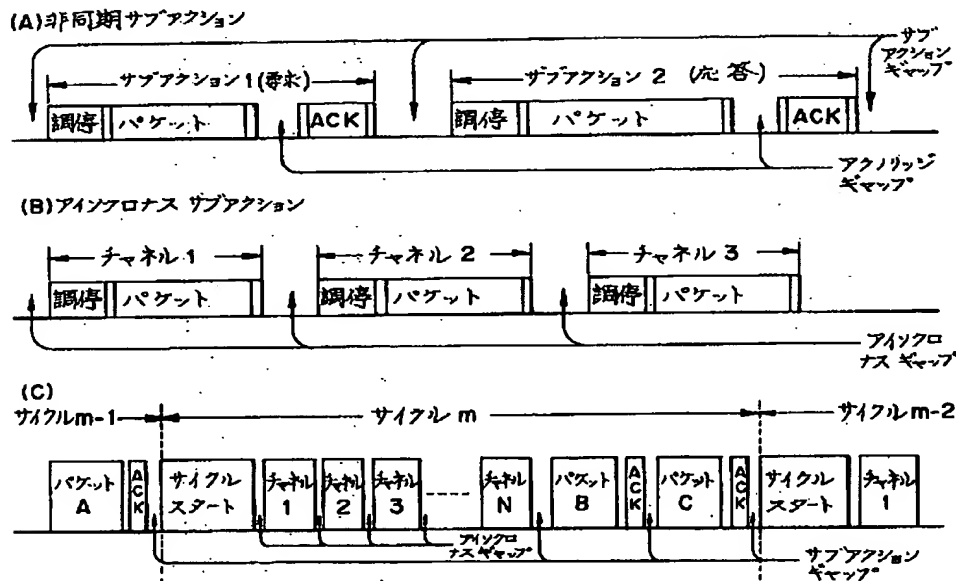
80 RAM (パケット記憶手段)  
 90、92、94 バス (第1のバス)  
 96、98 バス (第2のバス)  
 100、102、104、105、  
 106、107、108、109 バス (第3のバス)  
 110 バス (第4のバス)  
 120 データ転送制御装置  
 122 PHYチップ  
 124 アプリケーション層のデバイス  
 130 バス監視回路  
 132 直列・並列変換回路  
 142 パケット診断回路  
 160 パケット整形回路  
 162 TAG生成回路  
 164 ヘッダ&トレイラー生成回路  
 166 エラーチェック回路  
 167 シーケンサ

\*

\* 168 バッファ  
 170 セレクタ  
 180 パケット分離回路  
 182 TAG判別回路  
 184 ポインタ更新回路  
 188 アドレス発生回路  
 190 アクセス要求実行回路  
 192 アドレス要求発生回路  
 280 パケット結合回路  
 10 284 ポインタ更新回路  
 285 データポインタ取得回路  
 286 tcode判別回路  
 290 アクセス要求実行回路  
 292 アドレス要求発生回路  
 294 ACK書き込み要求発生回路  
 296 ACK書き込みデータ&アドレス発生回路

【図1】

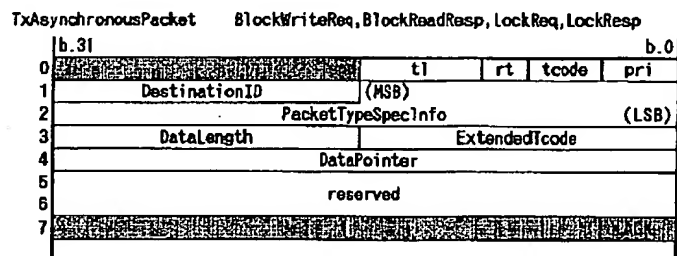
【図7】



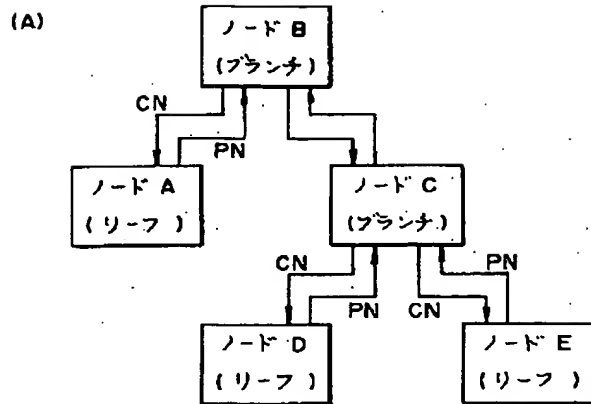
【図16】

TAG (DTAG)	意味
00	ヘッダ
01	トレイラー
10	データ
11	スタート

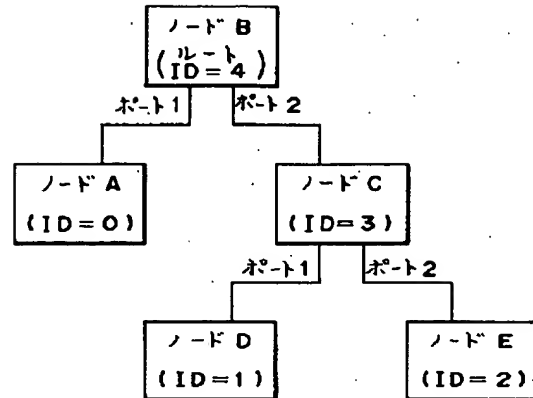
【図25】



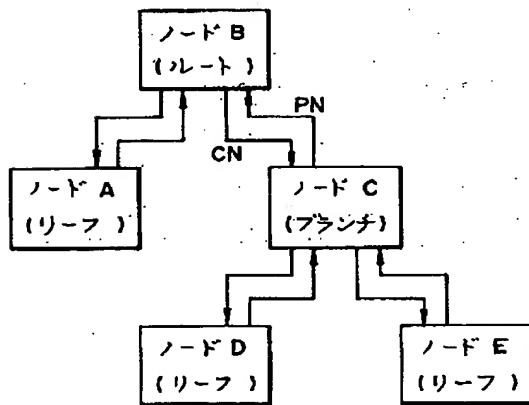
【図2】



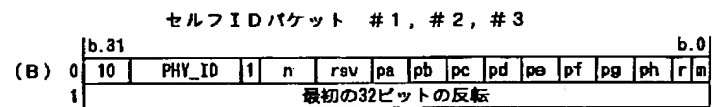
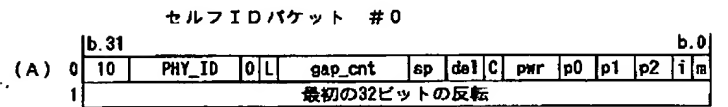
【図3】



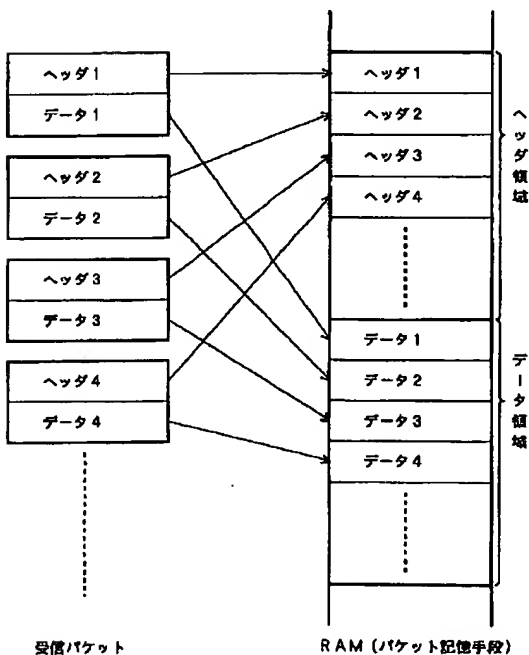
(B)



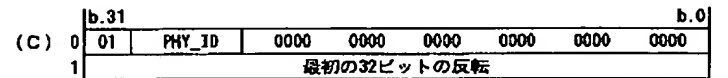
【図4】



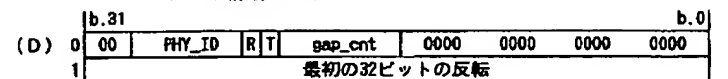
【図13】



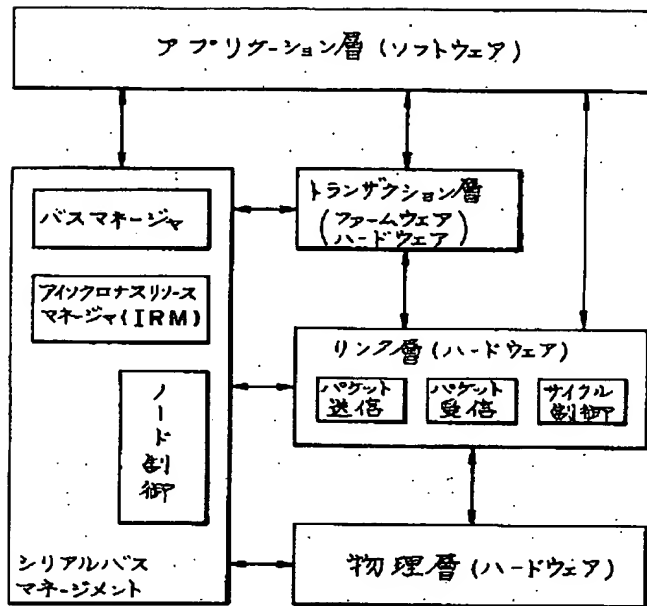
リンクオンパケット



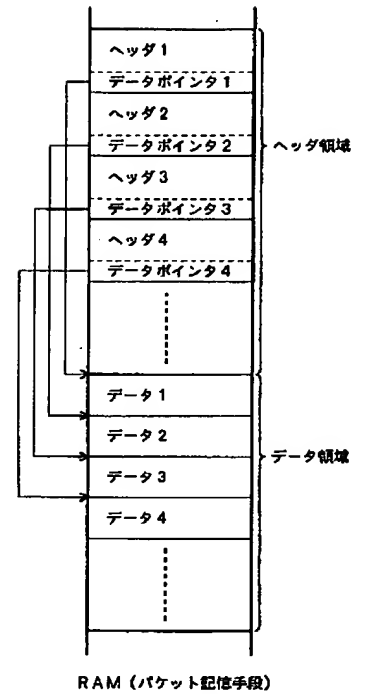
PHY構成パケット



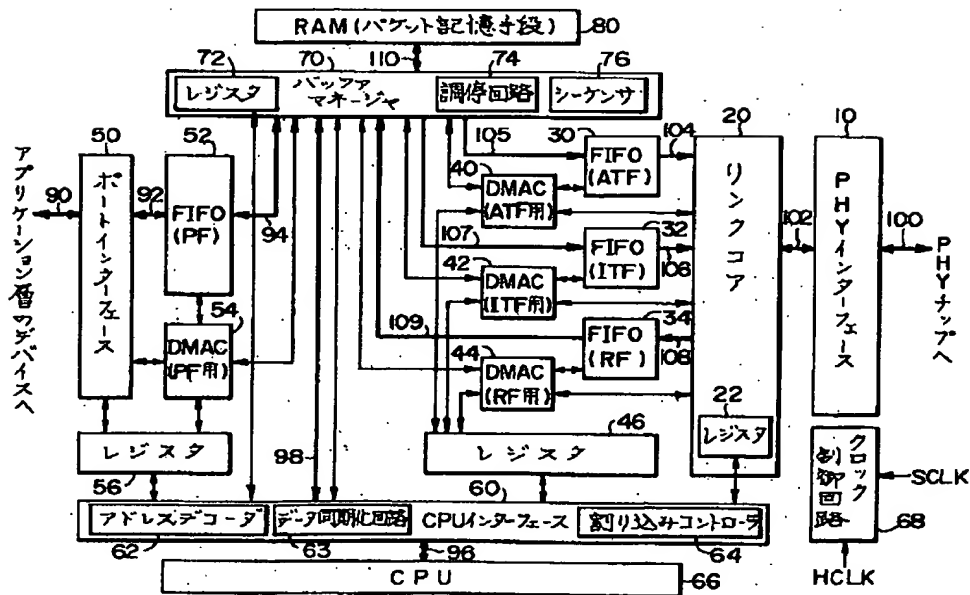
【図5】



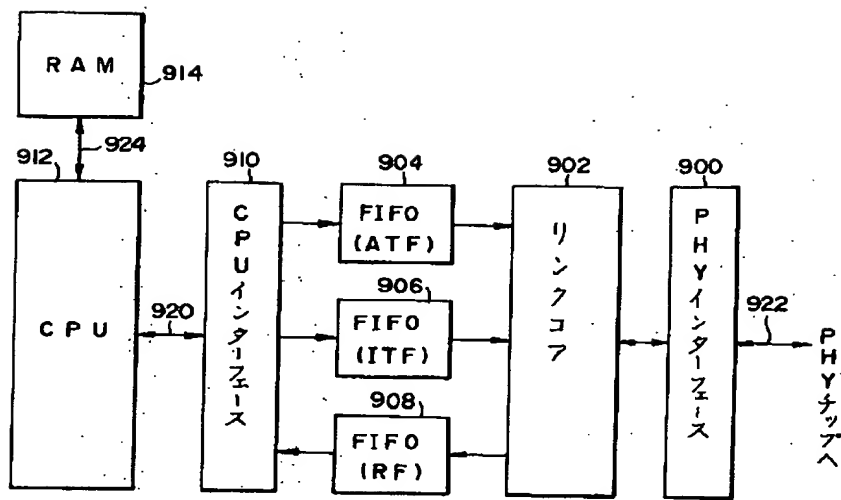
【図15】



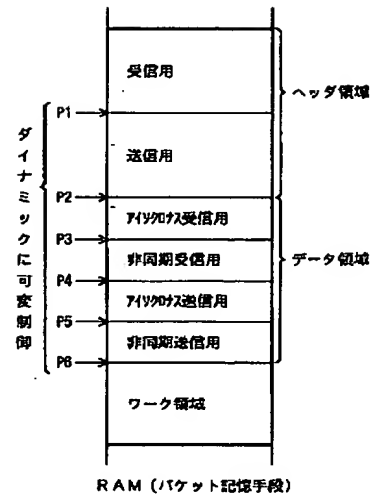
【図6】



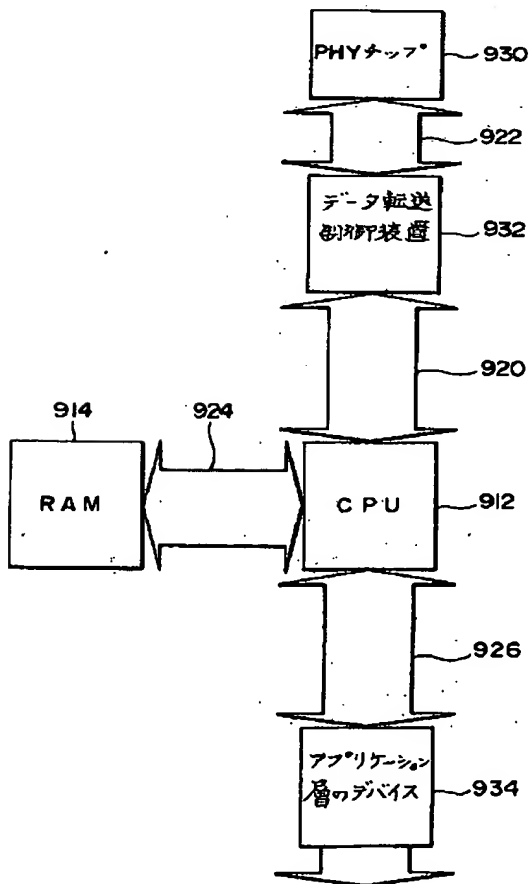
【図 8】



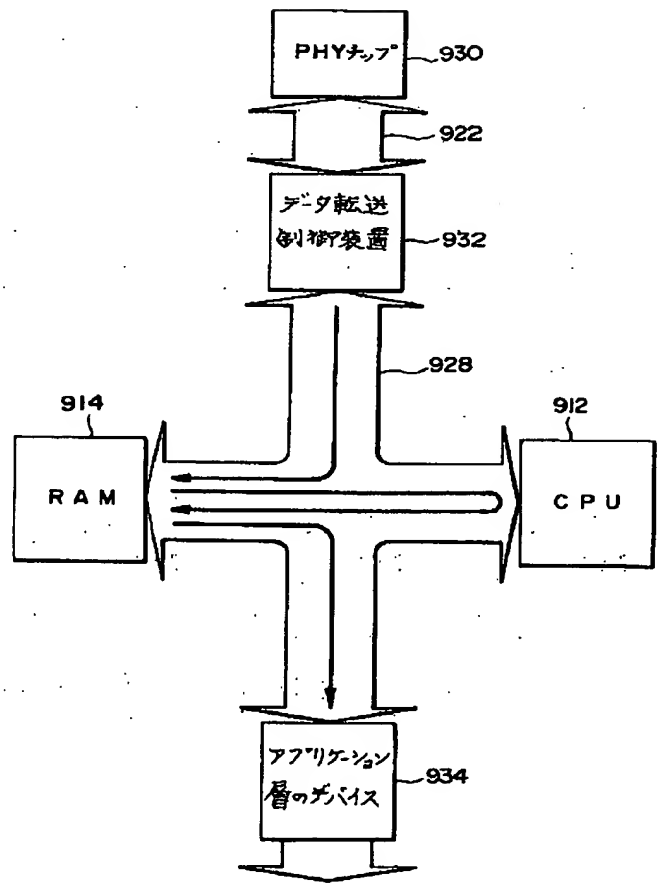
【図 18】



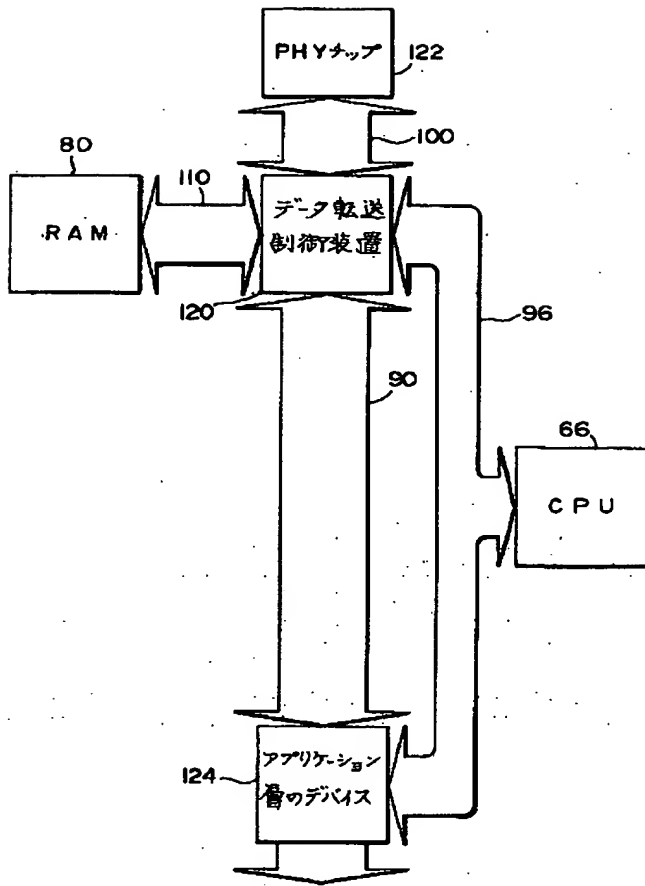
【図 9】



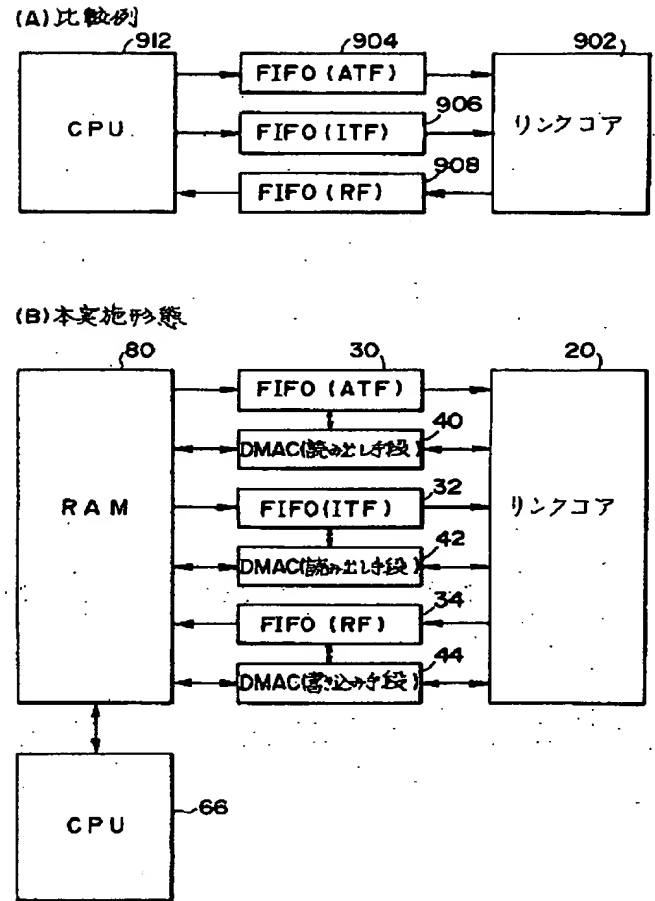
【図 10】



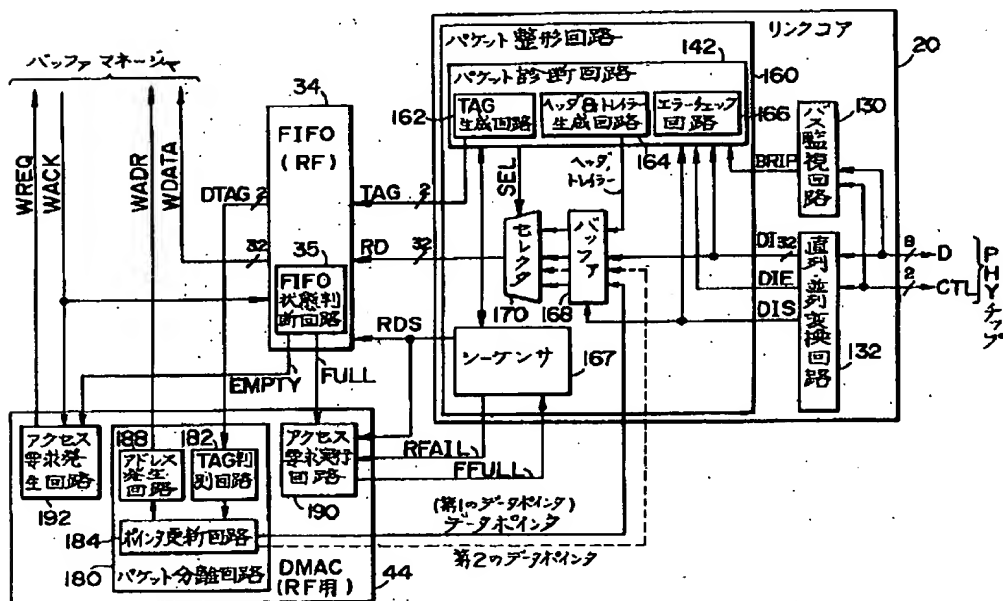
【図11】



【図12】

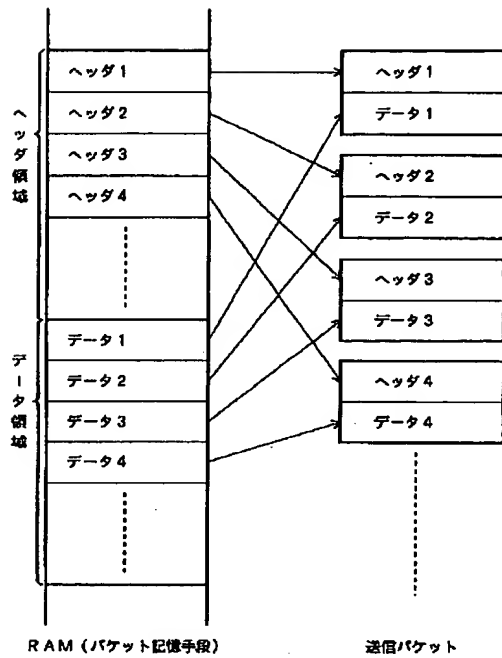


【図19】

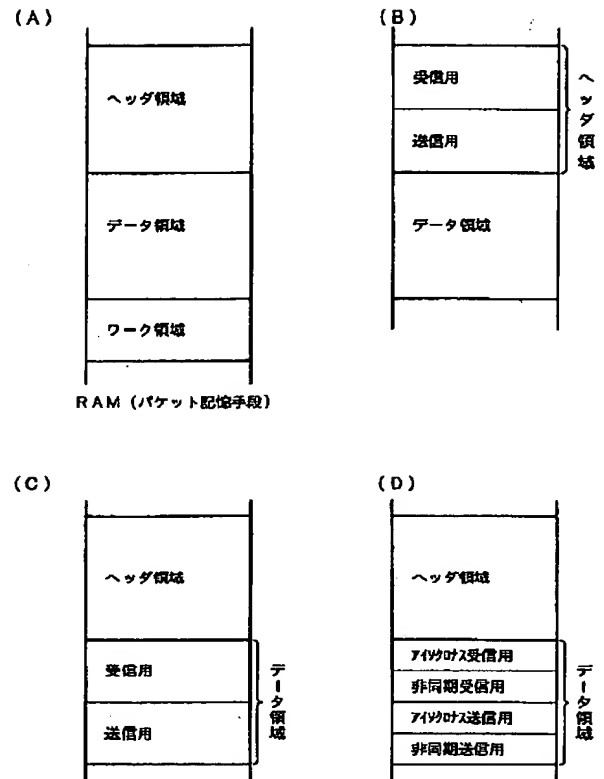




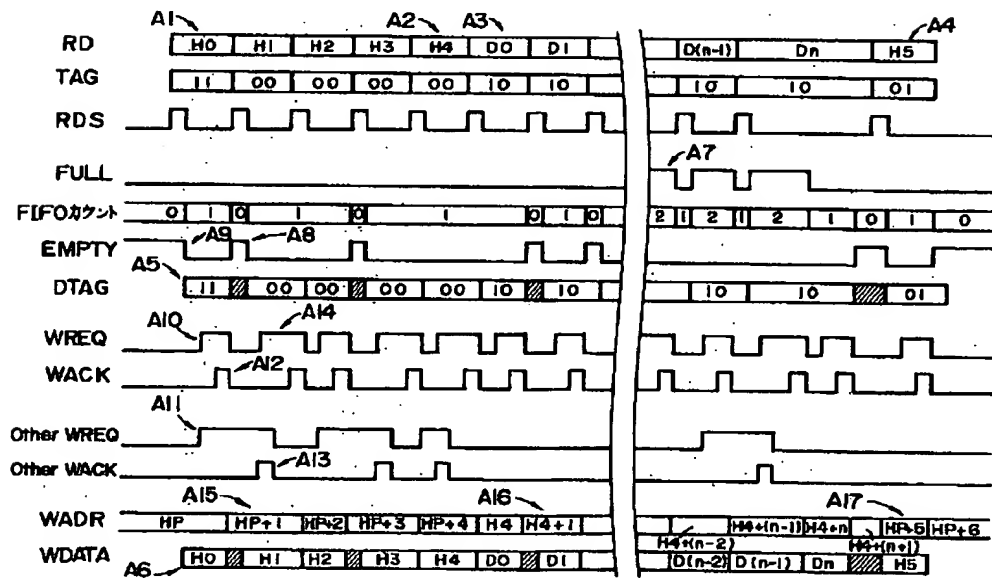
【図 14】



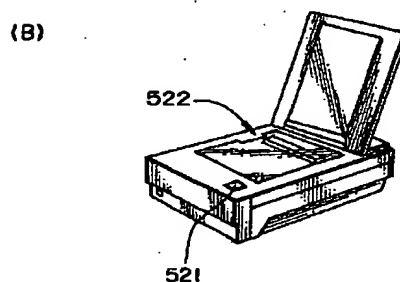
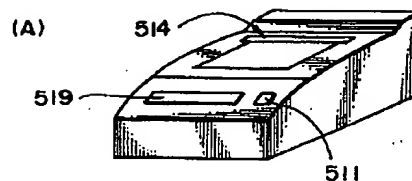
【図 17】



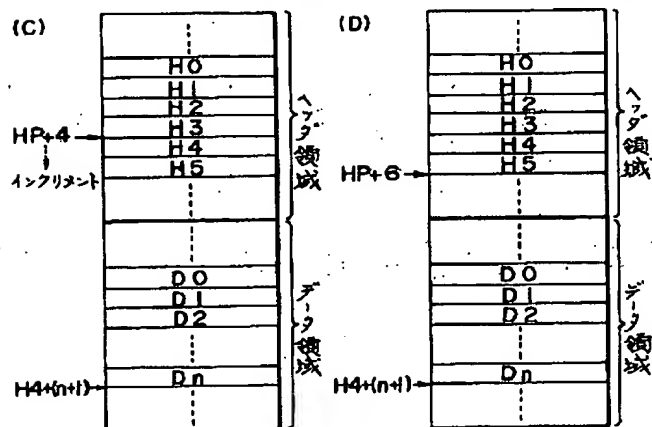
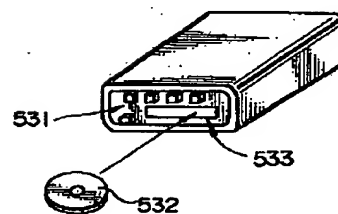
【図 20】



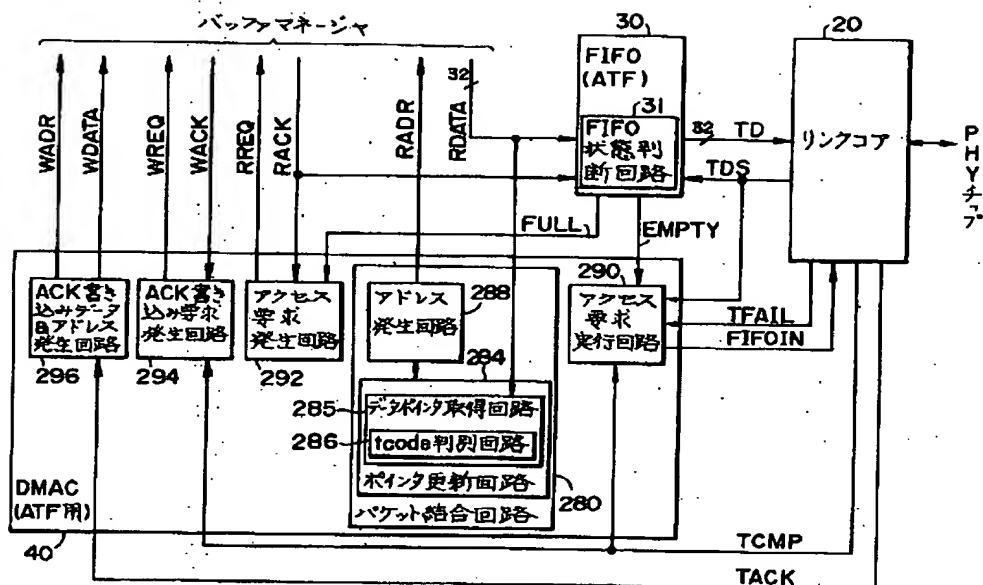
【图 27】



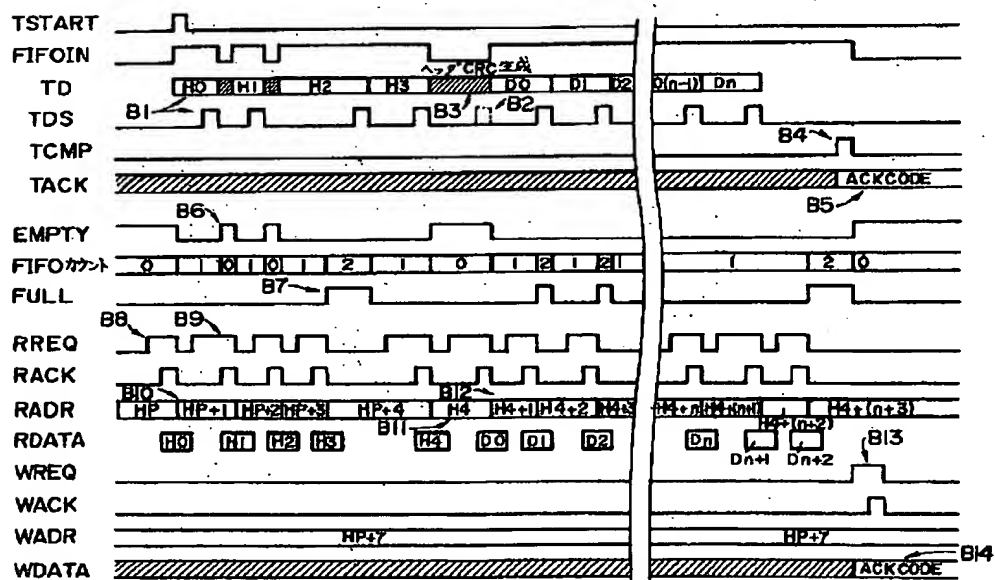
(C)



【図 23】



【图 24】



【図26】

